

EXHIBIT 13

**IN THE UNITED STATES DISTRICT COURT
FOR THE WESTERN DISTRICT OF TEXAS
WACO DIVISION**

WSOU INVESTMENTS, LLC D/B/A	§	
BRAZOS LICENSING AND	§	Case No. 6:20-cv-00957-ADA
DEVELOPMENT,	§	
<i>Plaintiff,</i>	§	JURY TRIAL DEMANDED
	§	
v.	§	
	§	
ONEPLUS TECHNOLOGY	§	
(SHENZHEN) CO., LTD.,	§	
<i>Defendant.</i>	§	

**PLAINTIFF'S AMENDED FINAL DISCLOSURES OF
PRELIMINARY INFRINGEMENT CONTENTIONS**

Pursuant to the Court's Order Governing Proceeding – Patent Case (“Order Governing Proceeding”), Plaintiff WSOU Investments, LLC d/b/a Brazos Licensing and Development (“WSOU”) hereby provides its Amended Final Infringements Contentions to defendant OnePlus Technology (Shenzhen) Co., Ltd. (“OnePlus” or “Defendant”) for U.S. Patent No. 8,712,708 (the “708 Patent”).

WSOU makes this disclosure based on the information presently available to it. Discovery in this case has not started, and WSOU reserves its right to amend or supplement these disclosures as permitted by the Federal Rules of Civil Procedure, by the local rules of the Western District of Texas, and by order of the Court, including the Court's Order Governing Proceedings.

For each Asserted Claim, Plaintiff identifies the following Accused Instrumentalities of which it is currently aware. The identification of Accused Instrumentalities is based on Plaintiff's research and analysis to date, without the benefit of discovery from the Defendant. Plaintiff reserves the right to add, delete, substitute or otherwise amend this list of Accused

Instrumentalities based on discovery or other circumstances, in a manner consistent with the Federal Rules of Civil Procedures, local rules, and standing orders.

The Accused Instrumentalities include, without limitation, the following:

- OnePlus Mobile Phones (such as OnePlus 8, OnePlus 7 Pro, OnePlus 8 Pro, OnePlus 8T, OnePlus 9, OnePlus 9 Pro).
- All past, current and future OnePlus products and services that operate in the same or substantially similar manner as the specifically identified products and services above and described in Exhibit 1.
- All past, current and future OnePlus products and services that have the same or substantially similar features as the specifically identified products and services above and described in Exhibit 1.

Plaintiff's infringement contentions apply to the Accused Instrumentalities as well as all other past, current and future hardware and software products and services developed, made, used, offered for sale, sold, imported, and provided by OnePlus that contain or makes use of the Patented Technology.¹

Based upon currently available information, WSOU asserts that OnePlus has infringed and/or continues to infringe the patent and claims identified in the attached claim charts (the "Asserted Claims" of the "Patent-in-Suit"). Infringement claim charts evidencing the correspondence between (i) the elements of the Asserted Claims, and (ii) the corresponding items of the accused products are attached hereto. Further, Exhibit 1, which is attached hereto and incorporated by reference, is an exemplary infringement claim chart identifying specifically where

¹ "Patented Technology" means all technologies described in the claims of the Patent-in-Suit.

each limitation of each Asserted Claim is found within each Accused Instrumentality or practiced by each Accused Instrumentality.

Plaintiff asserts that Defendant has directly infringed and continues to directly infringe the Asserted Claims literally through the Accused Instrumentalities by making, using, offering for sale, and/or selling, or importing into the United States the Accused Instrumentalities. To the extent that Defendant alleges that one or more limitations of the Asserted Claims are not literally found in the Accused Instrumentalities, Plaintiff alleges that such limitations are found in or practiced by the Accused Instrumentalities under the doctrine of equivalents. Any differences alleged to exist between any of the Asserted Claims and any of the Accused Instrumentalities are insubstantial and that each Accused Instrumentality also meets each limitation under the doctrine of equivalents as the identified features of the Accused Instrumentality performs substantially the same function in substantially the same way to achieve substantially the same result as the corresponding claim limitation. WSOU reserves the right to assert infringement solely under the doctrine of equivalents with respect to any particular claim element(s), if warranted by discovery, further analysis, and/or claim constructions in this case.

<u>SM8150</u> <u>(OnePlus 7 Pro)</u>
<pre>#define PSY_PROP(psy, prop, val) (power_supply_get_property(psy, \ POWER_SUPPLY_PROP_##prop, val)) #define _MPSY_PROP(prop, val) (power_supply_get_property(main_battery, \ prop, val)) #define MPSY_PROP(prop, val) _MPSY_PROP(POWER_SUPPLY_PROP_##prop, val)</pre>
Source
<pre>101 static ssize_t power_supply_show_property(struct device *dev, 102 struct device_attribute *attr, 103 char *buf) { 104 ssize_t ret = 0; 105 struct power_supply *psy = dev_get_drvdata(dev); 106 const ptrdiff_t off = attr - power_supply_attrs; 107 union power_supply_propval value;</pre>

Source
<pre> 230 /* Must be in the same order as POWER_SUPPLY_PROP_* */ 231 static struct device_attribute power_supply_attrs[] = { 232 /* Properties of type 'int' */ 233 POWER_SUPPLY_ATTR(status), 234 POWER_SUPPLY_ATTR(set_allow_read_extern_fg_iic), 235 POWER_SUPPLY_ATTR(cc_to_cv_point), 236 POWER_SUPPLY_ATTR(chg_protect_status), </pre>
Source
<pre> 262 POWER_SUPPLY_ATTR(voltage_max), 263 POWER_SUPPLY_ATTR(voltage_min), 264 POWER_SUPPLY_ATTR(voltage_max_design), 265 POWER_SUPPLY_ATTR(voltage_min_design), 266 POWER_SUPPLY_ATTR(voltage_now), 267 POWER_SUPPLY_ATTR(voltage_avg), 268 POWER_SUPPLY_ATTR(voltage_ocv), 269 POWER_SUPPLY_ATTR(voltage_boot), </pre>
<pre> 285 POWER_SUPPLY_ATTR(constant_charge_voltage), 286 POWER_SUPPLY_ATTR(constant_charge_voltage_max), </pre>
Source
<pre> 270 POWER_SUPPLY_ATTR(current_max), 271 POWER_SUPPLY_ATTR(current_now), 272 POWER_SUPPLY_ATTR(current_avg), 273 POWER_SUPPLY_ATTR(current_boot), </pre>
<pre> 283 POWER_SUPPLY_ATTR(constant_charge_current), 284 POWER_SUPPLY_ATTR(constant_charge_current_max), </pre>
<pre> 314 POWER_SUPPLY_ATTR(precache_current), 315 POWER_SUPPLY_ATTR(charge_term_current), </pre>
<pre> 333 POWER_SUPPLY_ATTR(input_current_max), 334 POWER_SUPPLY_ATTR(input_current_trim), 335 POWER_SUPPLY_ATTR(input_current_settled), </pre>
Source
<pre> 276 POWER_SUPPLY_ATTR(charge_full_design), 277 POWER_SUPPLY_ATTR(charge_empty_design), 278 POWER_SUPPLY_ATTR(charge_full), 279 POWER_SUPPLY_ATTR(charge_empty), 280 POWER_SUPPLY_ATTR(charge_now), 281 POWER_SUPPLY_ATTR(charge_avg), 282 POWER_SUPPLY_ATTR(charge_counter), 283 POWER_SUPPLY_ATTR(constant_charge_current), 284 POWER_SUPPLY_ATTR(constant_charge_current_max), 285 POWER_SUPPLY_ATTR(constant_charge_voltage), 286 POWER_SUPPLY_ATTR(constant_charge_voltage_max), 287 POWER_SUPPLY_ATTR(charge_control_limit), 288 POWER_SUPPLY_ATTR(charge_control_limit_max), </pre>
Source

```

371 static int axp288_charger_usb_set_property(struct power_supply *psy,
372                                         enum power_supply_property psp,
373                                         const union power_supply_propval *val)
374 {
375     struct axp288_chrg_info *info = power_supply_get_drvdata(psy);
376     int ret = 0;
377     int scaled_val;
378
379     mutex_lock(&info->lock);
380
381     switch (psp) {
382     case POWER_SUPPLY_PROP_CONSTANT_CHARGE_CURRENT:
383         scaled_val = min(val->intval, info->max_cc);
384         scaled_val = DIV_ROUND_CLOSEST(scaled_val, 1000);
385         ret = axp288_charger_set_cc(info, scaled_val);
386         if (ret < 0)
387             dev_warn(&info->pdev->dev, "set charge current failed\n");
388         break;
389     case POWER_SUPPLY_PROP_CONSTANT_CHARGE_VOLTAGE:
390         scaled_val = min(val->intval, info->max_cv);
391         scaled_val = DIV_ROUND_CLOSEST(scaled_val, 1000);
392         ret = axp288_charger_set_cv(info, scaled_val);
393         if (ret < 0)
394             dev_warn(&info->pdev->dev, "set charge voltage failed\n");
395         break;
396     default:
397         ret = -EINVAL;
398     }

```

[Source](#)

SM8250

(OnePlus 8, 8 Pro, 8T)

```

17 #define PSY_PROP(psy, prop, val) (power_supply_get_property(psy, \
18                                     POWER_SUPPLY_PROP_##prop, val))
19
20 #define _MPSY_PROP(prop, val) (power_supply_get_property(main_battery, \
21                                     prop, val))
22
23 #define MPSY_PROP(prop, val) _MPSY_PROP(POWER_SUPPLY_PROP_##prop, val)

```

Source: Exhibit G

```

144 static ssize_t power_supply_show_property(struct device *dev,
145                                         struct device_attribute *attr,
146                                         char *buf) {
147     ssize_t ret;
148     struct power_supply *psy = dev_get_drvdata(dev);
149     enum power_supply_property psp = attr - power_supply_attrs;
150     union power_supply_propval value;

```

Source: Exhibit E

```

298 /* Must be in the same order as POWER_SUPPLY_PROP_ */
299 static struct device_attribute power_supply_attrs[] = {
300     /* Properties of type 'int' */
301     POWER_SUPPLY_ATTR(status),
302     /* @bsp, 2018/07/13 Battery & Charging porting */
303     POWER_SUPPLY_ATTR(set_allow_read_extern_fg_iic),
304     POWER_SUPPLY_ATTR(cc_to_cv_point),
305     POWER_SUPPLY_ATTR(chg_protect_status),

```

Source: Exhibit E

```

332     POWER_SUPPLY_ATTR(voltage_max),
333     POWER_SUPPLY_ATTR(voltage_min),
334     POWER_SUPPLY_ATTR(voltage_max_design),
335     POWER_SUPPLY_ATTR(voltage_min_design),
336     POWER_SUPPLY_ATTR(voltage_now),
337     POWER_SUPPLY_ATTR(voltage_avg),
338     POWER_SUPPLY_ATTR(voltage_ocv),
339     POWER_SUPPLY_ATTR(voltage_boot),
340     POWER_SUPPLY_ATTR(constant_charge_voltage),
341     POWER_SUPPLY_ATTR(constant_charge_voltage_max),
342     POWER_SUPPLY_ATTR(voltage_max_limit),
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
589
590
591
592
593
594
595
596
597
598
599
599
600
601
602
603
604
605
606
607
608
609
609
610
611
612
613
614
615
616
617
618
619
619
620
621
622
623
624
625
626
627
628
629
629
630
631
632
633
634
635
636
637
638
639
639
640
641
642
643
644
645
646
647
647
648
649
649
650
651
652
653
654
655
656
657
657
658
659
659
660
661
662
663
664
665
666
666
667
668
668
669
669
670
671
672
673
674
675
676
676
677
678
678
679
679
680
681
682
683
684
685
686
686
687
687
688
688
689
689
690
691
692
693
694
695
696
697
698
699
699
700
701
702
703
704
705
706
707
708
709
709
710
711
712
713
714
715
716
717
718
719
719
720
721
722
723
724
725
726
727
728
729
729
730
731
732
733
734
735
736
737
738
739
739
740
741
742
743
744
745
746
747
747
748
749
749
750
751
752
753
754
755
756
757
758
759
759
760
761
762
763
764
765
766
767
768
769
769
770
771
772
773
774
775
776
777
778
779
779
780
781
782
783
784
785
786
787
787
788
789
789
790
791
792
793
794
795
796
797
798
799
799
800
801
802
803
804
805
806
807
808
809
809
810
811
812
813
814
815
816
817
818
819
819
820
821
822
823
824
825
826
827
828
829
829
830
831
832
833
834
835
836
837
838
839
839
840
841
842
843
844
845
846
847
847
848
849
849
850
851
852
853
854
855
856
857
858
859
859
860
861
862
863
864
865
866
867
867
868
869
869
870
871
872
873
874
875
876
877
877
878
879
879
880
881
882
883
884
885
886
887
887
888
889
889
890
891
892
893
894
895
896
897
898
899
899
900
901
902
903
904
905
906
907
908
909
909
910
911
912
913
914
915
916
917
917
918
919
919
920
921
922
923
924
925
926
927
927
928
929
929
930
931
932
933
934
935
936
937
937
938
939
939
940
941
942
943
944
945
946
947
947
948
949
949
950
951
952
953
954
955
956
957
957
958
959
959
960
961
962
963
964
965
966
966
967
968
968
969
969
970
971
972
973
974
975
976
977
977
978
979
979
980
981
982
983
984
985
986
987
987
988
989
989
990
991
992
993
994
995
996
997
998
999
999
1000
1000
1001
1002
1003
1004
1005
1006
1007
1008
1009
1009
1010
1011
1012
1013
1014
1015
1015
1016
1017
1017
1018
1018
1019
1020
1021
1022
1023
1024
1025
1025
1026
1027
1027
1028
1028
1029
1030
1031
1032
1033
1034
1035
1035
1036
1037
1037
1038
1038
1039
1040
1041
1042
1043
1044
1045
1045
1046
1047
1047
1048
1048
1049
1050
1051
1052
1053
1054
1055
1055
1056
1057
1057
1058
1058
1059
1060
1061
1062
1063
1064
1065
1065
1066
1067
1067
1068
1068
1069
1070
1071
1072
1073
1074
1075
1075
1076
1077
1077
1078
1078
1079
1080
1081
1082
1083
1084
1085
1085
1086
1087
1087
1088
1088
1089
1090
1091
1092
1093
1094
1095
1095
1096
1097
1097
1098
1098
1099
1099
1100
1101
1102
1103
1104
1105
1105
1106
1107
1107
1108
1108
1109
1110
1111
1112
1113
1114
1115
1115
1116
1117
1117
1118
1118
1119
1120
1121
1122
1123
1124
1125
1125
1126
1127
1127
1128
1128
1129
1130
1131
1132
1133
1134
1135
1135
1136
1137
1137
1138
1138
1139
1140
1141
1142
1143
1144
1145
1145
1146
1147
1147
1148
1148
1149
1150
1151
1152
1153
1154
1155
1155
1156
1157
1157
1158
1158
1159
1160
1161
1162
1163
1164
1165
1165
1166
1167
1167
1168
1168
1169
1170
1171
1172
1173
1174
1175
1175
1176
1177
1177
1178
1178
1179
1180
1181
1182
1183
1184
1185
1185
1186
1187
1187
1188
1188
1189
1190
1191
1192
1193
1194
1195
1195
1196
1197
1197
1198
1198
1199
1200
1201
1202
1203
1204
1205
1205
1206
1207
1207
1208
1208
1209
1210
1211
1212
1213
1214
1215
1215
1216
1217
1217
1218
1218
1219
1220
1221
1222
1223
1224
1225
1225
1226
1227
1227
1228
1228
1229
1230
1231
1232
1233
1234
1235
1235
1236
1237
1237
1238
1238
1239
1240
1241
1242
1243
1244
1245
1245
1246
1247
1247
1248
1248
1249
1250
1251
1252
1253
1254
1255
1255
1256
1257
1257
1258
1258
1259
1260
1261
1262
1263
1264
1265
1265
1266
1267
1267
1268
1268
1269
1270
1271
1272
1273
1274
1275
1275
1276
1277
1277
1278
1278
1279
1280
1281
1282
1283
1284
1285
1285
1286
1287
1287
1288
1288
1289
1290
1291
1292
1293
1294
1295
1295
1296
1297
1297
1298
1298
1299
1300
1301
1302
1303
1304
1305
1305
1306
1307
1307
1308
1308
1309
1310
1311
1312
1313
1314
1315
1315
1316
1317
1317
1318
1318
1319
1320
1321
1322
1323
1324
1325
1325
1326
1327
1327
1328
1328
1329
1330
1331
1332
1333
1334
1335
1335
1336
1337
1337
1338
1338
1339
1340
1341
1342
1343
1344
1345
1345
1346
1347
1347
1348
1348
1349
1350
1351
1352
1353
1354
1355
1355
1356
1357
1357
1358
1358
1359
1360
1361
1362
1363
1364
1365
1365
1366
1367
1367
1368
1368
1369
1370
1371
1372
1373
1374
1375
1375
1376
1377
1377
1378
1378
1379
1380
1381
1382
1383
1384
1385
1385
1386
1387
1387
1388
1388
1389
1390
1391
1392
1393
1394
1395
1395
1396
1397
1397
1398
1398
1399
1400
1401
1402
1403
1404
1405
1405
1406
1407
1407
1408
1408
1409
1410
1411
1412
1413
1414
1415
1415
1416
1417
1417
1418
1418
1419
1420
1421
1422
1423
1424
1425
1425
1426
1427
1427
1428
1428
1429
1430
1431
1432
1433
1434
1435
1435
1436
1437
1437
1438
1438
1439
1440
1441
1442
1443
1444
1445
1445
1446
1447
1447
1448
1448
1449
1450
1451
1452
1453
1454
1455
1455
1456
1457
1457
1458
1458
1459
1460
1461
1462
1463
1464
1465
1465
1466
1467
1467
1468
1468
1469
1470
1471
1472
1473
1474
1475
1475
1476
1477
1477
1478
1478
1479
1480
1481
1482
1483
1484
1485
1485
1486
1487
1487
1488
1488
1489
1490
1491
1492
1493
1494
1495
1495
1496
1497
1497
1498
1498
1499
1500
1501
1502
1503
1504
1505
1505
1506
1507
1507
1508
1508
1509
1510
1511
1512
1513
1514
1515
1515
1516
1517
1517
1518
1518
1519
1520
1521
1522
1523
1524
1525
1525
1526
1527
1527
1528
1528
1529
1530
1531
1532
1533
1534
1535
1535
1536
1537
1537
1538
1538
1539
1540
1541
1542
1543
1544
1545
1545
1546
1547
1547
1548
1548
1549
1550
1551
1552
1553
1554
1555
1555
1556
1557
1557
1558
1558
1559
1560
1561
1562
1563
1564
1565
1565
1566
1567
1567
1568
1568
1569
1570
1571
1572
1573
1574
1575
1575
1576
1577
1577
1578
1578
1579
1580
1581
1582
1583
1584
1585
1585
1586
1587
1587
1588
1588
1589
1590
1591
1592
1593
1594
1595
1595
1596
1597
1597
1598
1598
1599
1600
1601
1602
1603
1604
1605
1605
1606
1607
1607
1608
1608
1609
1610
1611
1612
1613
1614
1615
1615
1616
1617
1617
1618
1618
1619
1620
1621
1622
1623
1624
1625
1625
1626
1627
1627
1628
1628
1629
1630
1631
1632
1633
1634
1635
1635
1636
1637
1637
1638
1638
1639
1640
1641
1642
1643
1644
1645
1645
1646
1647
1647
1648
1648
1649
1650
1651
1652
1653
1654
1655
1655
1656
1657
1657
1658
1658
1659
1660
1661
1662
1663
1664
1665
1665
1666
1667
1667
1668
1668
1669
1670
1671
1672
1673
1674
1675
1675
1676
1677
1677
1678
1678
1679
1680
1681
1682
1683
1684
1685
1685
1686
1687
1687
1688
1688
1689
1690
1691
1692
1693
1694
1695
1695
1696
1697
1697
1698
1698
1699
1700
1701
1702
1703
1704
1705
1705
1706
1707
1707
1708
1708
1709
1710
1711
1712
1713
1714
1715
1715
1716
1717
1717
1718
1718
1719
1720
1721
1722
1723
1724
1725
1725
1726
1727
1727
1728
1728
1729
1730
1731
1732
1733
1734
1735
1735
1736
1737
1737
1738
1738
1739
1740
1741
1742
1743
1744
1745
1745
1746
1747
1747
1748
1748
1749
1750
1751
1752
1753
1754
1755
1755
1756
1757
1757
1758
1758
1759
1760
1761
1762
1763
1764
1765
1765
1766
1767
1767
1768
1768
1769
1770
1771
1772
1773
1774
1775
1775
1776
1777
1777
1778
1778
1779
1780
1781
1782
1783
1784
1785
1785
1786
1787
1787
1788
1788
1789
1790
1791
1792
1793
1794
1795
1795
1796
1797
1797
1798
1798
1799
1800
1801
1802
1803
1804
1805
1805
1806
1807
1807
1808
1808
1809
1810
1811
1812
1813
1814
1815
1815
1816
1817
1817
1818
1818
1819
1820
1821
1822
1823
1824
1825
1825
1826
1827
1827
1828
1828
1829
1830
1831
1832
1833
1834
1835
1835
1836
1837
1837
1838
1838
1839
1840
1841
1842
1843
1844
1845
1845
1846
1847
1847
1848
1848
1849
1850
1851
1852
1853
1854
1855
1855
1856
1857
1857
1858
1858
1859
1860
1861
1862
1863
1864
1865
1865
1866
1867
1867
1868
1868
1869
1870
1871
1872
1873
1874
1875
1875
1876
1877
1877
1878
1878
1879
1880
1881
1882
1883
1884
1885
1885
1886
1887
1887
1888
1888
1889
1890
1891
1892
1893
1894
1895
1895
1896
1897
1897
1898
1898
1899
1900
1901
1902
1903
1904
1905
1905
1906
1907
1907
1908
1908
1909
1910
1911
1912
1913
1914
1915
1915
1916
1917
1917
1918
1918
1919
1920
1921
1922
1923
1924
1925
1925
1926
1927
1927
1928
1928
1929
1930
1931
1932
1933
1934
1935
1935
1936
1937
1937
1938
1938
1939
1940
1941
1942
1943
1944
1945
1945
1946
1947
1947
1948
1948
1949
1950
1951
1952
1953
1954
1955
1955
1956
1957
1957
1958
1958
1959
1960
1961
1962
1963
1964
1965
1965
1966
1967
1967
1968
1968
1969
1970
1971
1972
1973
1974
1975
1975
1976
1977
1977
1978
1978
1979
1980
1981
1982
1983
1984
1985
1985
1986
1987
1987
1988
1988
1989
1990
1991
1992
1993
1994
1995
1995
1996
1997
1997
1998
1998
1999
2000
2001
2002
2003
2004
2005
2005
2006
2007
2007
2008
2008
2009
2010
2011
2012
2013
2014
2015
2015
2016
2017
2017
2018
2018
2019
2020
2021
2022
2023
2024
2025
2025
2026
2027
2027
2028
2028
2029
2030
2031
2032
2033
2034
2035
2035
2036
2037
2037
2038
2038
2039
2040
2041
2042
2043
2044
2045
2045
2046
2047
2047
2048
2048
2049
2050
2051
2052
2053
2054
2055
2055
2056
2057
2057
2058
2058
2059
2060
2061
2062
2063
2064
2065
2065
2066
2067
2067
2068
2068
2069
2070
2071
2072
2073
2074
2075
2075
2076
2077
2077
2078
2078
2079
2080
2081
2082
2083
2084
2085
2085
2086
2087
2087
2088
2088
2089
2090
2091
2092
2093
2094
2095
2095
2096
2097
2097
2098
2098
2099
2100
2101
2102
2103
2104
2105
2105
2106
2107
2107
2108
2108
2109
2110
2111
2112
2113
2114
2115
2115
2116
2117
2117
2118
2118
2119
2120
2121
2122
2123
2124
2125
2125
2126
2127
2127
2128
2128
2129
2130
2131
2132
2133
2134
2135
2135
2136
2137
2137
2138
2138
2139
2140
2141
2142
2143
2144
2145
2145
2146
2147
2147
2148
2148
2149
2150
2151
2152
2153
2154
2155
2155
2156
2157
2157
2158
2158
2159
2160
2161
2162
2163
2164
2165
2165
2166
2167
2167
2168
2168
2169
2170
2171
2172
2173
2174
2175
2175
2176
2177
2177
2178
2178
2179
2180
2181
2182
2183
2184
2185
2185
2186
2187
2187
2188
2188
2189
2190
2191
2192
2193
2194
2195
2195
2196
2197
2197
2198
2198
2199
2200
2201
2202
2203
2204
2205
2205
2206
2207
2207
2208
2208
2209
2210
2211
2212
2213
2214
2215
2215
2216
2217
2217
2218
2218
2219
2220
2221
2222
2223
2224
2225
2225
2226
2227
2227
2228
2228
2229
2230
2231
2232
2233
2234
2235
2235
2236
2237
2237
2238
2238
2239
2240
2241
2242
2243
2244
2245
2245
2246
2247
2247
2248
2248
2249
2250
2251
2252
2253
2254
2255
2255
2256
2257
2257
2258
2258
2259
2260
2261
2262
2263
2264
2265
2265
2266
2267
2267
2268
2268
2269
2270
2271
2272
2273
2
```

Source: Exhibit H	
	<u>SM8350</u> <u>(OnePlus 9, 9 Pro)</u>
	<pre>#define PSY_PROP(psy, prop, val) (power_supply_get_property(psy, \ POWER_SUPPLY_PROP_##prop, val)) #define _MPSY_PROP(prop, val) (power_supply_get_property(main_battery, \ prop, val)) #define MPSY_PROP(prop, val) _MPSY_PROP(POWER_SUPPLY_PROP_##prop, val)</pre>
	Source
	<pre>266 static ssize_t power_supply_show_property(struct device *dev, 267 struct device_attribute *attr, 268 char *buf) { 269 ssize_t ret; 270 struct power_supply *psy = dev_get_drvdata(dev); 271 struct power_supply_attr *ps_attr = to_ps_attr(attr); 272 enum power_supply_property psp = dev_attr_psp(attr); 273 union power_supply_propval value;</pre>
	Source
	<pre>135 static struct power_supply_attr power_supply_attrs[] = { 136 /* Properties of type `int` */ 137 POWER_SUPPLY_ENUM_ATTR(STATUS), 138 POWER_SUPPLY_ENUM_ATTR(CHARGE_TYPE), 139 POWER_SUPPLY_ENUM_ATTR(HEALTH), 140 POWER_SUPPLY_ATTR(PRESENT), 141 POWER_SUPPLY_ATTR(ONLINE), 142 POWER_SUPPLY_ATTR(AUTHENTIC), 143 POWER_SUPPLY_ENUM_ATTR(TECHNOLOGY),</pre>
	Source
	<pre>145 POWER_SUPPLY_ATTR(VOLTAGE_MAX), 146 POWER_SUPPLY_ATTR(VOLTAGE_MIN), 147 POWER_SUPPLY_ATTR(VOLTAGE_MAX_DESIGN), 148 POWER_SUPPLY_ATTR(VOLTAGE_MIN_DESIGN), 149 POWER_SUPPLY_ATTR(VOLTAGE_NOW), 150 POWER_SUPPLY_ATTR(VOLTAGE_AVG), 151 POWER_SUPPLY_ATTR(VOLTAGE_OCV), 152 POWER_SUPPLY_ATTR(VOLTAGE_BOOT), 153 POWER_SUPPLY_ATTR(CONSTANT_CHARGE_VOLTAGE), 154 POWER_SUPPLY_ATTR(CONSTANT_CHARGE_VOLTAGE_MAX),</pre>
	Source
	<pre>155 POWER_SUPPLY_ATTR(CURRENT_MAX), 156 POWER_SUPPLY_ATTR(CURRENT_NOW), 157 POWER_SUPPLY_ATTR(CURRENT_AVG), 158 POWER_SUPPLY_ATTR(CURRENT_BOOT), 159 POWER_SUPPLY_ATTR(CONSTANT_CHARGE_CURRENT), 160 POWER_SUPPLY_ATTR(CONSTANT_CHARGE_CURRENT_MAX),</pre>
	Source

<pre> 203 POWER_SUPPLY_ATTR(PRECHARGE_CURRENT), 204 POWER_SUPPLY_ATTR(CHARGE_TERM_CURRENT), </pre>	Source
<pre> 159 POWER_SUPPLY_ATTR(CHARGE_FULL_DESIGN), 160 POWER_SUPPLY_ATTR(CHARGE_EMPTY_DESIGN), 161 POWER_SUPPLY_ATTR(CHARGE_FULL), 162 POWER_SUPPLY_ATTR(CHARGE_EMPTY), 163 POWER_SUPPLY_ATTR(CHARGE_NOW), 164 POWER_SUPPLY_ATTR(CHARGE_AVG), 165 POWER_SUPPLY_ATTR(CHARGE_COUNTER), 166 POWER_SUPPLY_ATTR(CONSTANT_CHARGE_CURRENT), 167 POWER_SUPPLY_ATTR(CONSTANT_CHARGE_CURRENT_MAX), 168 POWER_SUPPLY_ATTR(CONSTANT_CHARGE_VOLTAGE), 169 POWER_SUPPLY_ATTR(CONSTANT_CHARGE_VOLTAGE_MAX), 170 POWER_SUPPLY_ATTR(CHARGE_CONTROL_LIMIT), 171 POWER_SUPPLY_ATTR(CHARGE_CONTROL_LIMIT_MAX), </pre>	Source
<pre> 357 static int exp288_charger_usb_set_property(struct power_supply *psy, 358 enum power_supply_property psp, 359 const union power_supply_propval *val) 360 { 361 struct exp288_chrg_info *info = power_supply_get_drvdata(psy); 362 int ret = 0; 363 int scaled_val; 364 365 switch (psp) { 366 case POWER_SUPPLY_PROP_CONSTANT_CHARGE_CURRENT: 367 scaled_val = min(val->intval, info->max_cc); 368 scaled_val = DIV_ROUND_CLOSEST(scaled_val, 1000); 369 ret = exp288_charger_set_cc(info, scaled_val); 370 if (ret < 0) 371 dev_warn(&info->pdev->dev, "set charge current failed\n"); 372 break; 373 case POWER_SUPPLY_PROP_CONSTANT_CHARGE_VOLTAGE: 374 scaled_val = min(val->intval, info->max_cv); 375 scaled_val = DIV_ROUND_CLOSEST(scaled_val, 1000); 376 ret = exp288_charger_set_cv(info, scaled_val); 377 if (ret < 0) 378 dev_warn(&info->pdev->dev, "set charge voltage failed\n"); 379 break; 380 case POWER_SUPPLY_PROP_INPUT_CURRENT_LIMIT: 381 ret = exp288_charger_set_vbus_inlimt(info, val->intval); 382 if (ret < 0) 383 dev_warn(&info->pdev->dev, "set input current limit failed\n"); 384 break; 385 default: 386 ret = -EINVAL; 387 } </pre>	Source

Plaintiff further asserts that Defendant has indirectly infringed and continues to indirectly infringe by actively inducing infringement of one or more of the claims of the Asserted Patent through the Accused Instrumentalities. Plaintiff also asserts that these third-parties directly infringe at least one or more of the claims of the Asserted Patent through the manufacture, use, sale, offer to sell, or importation of the Accused Instrumentalities.

For example, Defendant has actively induced infringement by encouraging the use of the Accused Instrumentalities in ways that infringe each Asserted Claim, including, but not limited through providing instructions to its customers and partners to encourage and instruct the user or partner to utilize the accused product in an infringing manner. Defendant knew or should have known that such encouragement would induce infringement. Defendant has taken active steps with the specific intent to encourage and cause others to use each Accused Instrumentality in ways that infringe each Asserted Claim. Such active steps by Defendant with specific intent to induce infringement have included, among other things, advertising, promoting, marketing, making available for use, offering to sell, and/or selling the Accused Instrumentalities to others; encouraging and influencing others to import, offer to sell, and/or sell the Accused Instrumentalities; directing and instructing others to use the Accused Instrumentalities in infringing ways; and by providing the Accused Instrumentalities to others. OnePlus has performed the aforementioned active steps with the knowledge of the Asserted Patent at least as of the date when the complaint in this case was filed. OnePlus has known or should have known that the acts it has induced constitute infringement because, for instance, it has been aware that end users and resellers will purchase the Accused Instrumentalities will use them, resulting in direct infringement.

Further, for instance, the Accused Instrumentalities are known by Defendant to be especially made or especially adapted for use to infringe the Asserted Patent, and are not staple articles or commodity of commerce suitable for substantial non-infringing uses. Defendant contributes to the infringement of the Asserted Patent by making available for use, offering for sale, selling, and/or importing the Accused Instrumentalities to third parties, who use the Accused Instrumentalities and/or practice one or more claims of the Asserted Patent. Moreover, Defendant

has had notice of the Asserted Patent at least as of the filing of the Complaint in this case.

These Infringement Contentions, including Exhibit 1, are based upon publicly-available information, and Plaintiff's research and analysis to date. The Accused Instrumentalities involve confidential, proprietary designs that are not publicly available, and Defendant has not yet provided discovery. Discovery is ongoing, and Plaintiff anticipates that the subject matter of these infringement contentions will be the subject of expert discovery. Discovery will provide evidence of Defendant's infringement, may lead to the discovery of additional instances of infringement, and may also enable identification of additional claims that are infringed by Defendant. Plaintiff reserves the right to add, delete, substitute, or otherwise further amend these Infringement Contentions based on discovery or other circumstances, in a manner consistent with the Federal Rules of Civil Procedures, local rules, and standing orders. Plaintiff explicitly reserves the right to further modify and/or supplement these contentions with additional or different theories and/or additional or different evidence. Further, WSOU reserves the right to supplement or revise its infringement contentions and/or chart, including identification of additional asserted claims, based on, for example, new versions or variations of one or more of the Accused Instrumentalities that are later discovered.

PRIORITY DATE

Each of the Asserted Claims of the '708 Patent is entitled to a priority date of no later than Oct. 25, 2010. The subject matter described by the Asserted Claims, however, may have been conceived and reduced to practice prior to this priority date. WSOU also reserves the right to identify any portions of the file history as containing evidence of conception and reduction to practice. Plaintiff's research and analysis is ongoing and Plaintiff reserves the right to assert that the claims are entitled to a priority date that is earlier than the above date.

Dated: October 26, 2021

RESPECTFULLY SUBMITTED,

By: /s/ Jonathan K. Waldrop
Jonathan K. Waldrop (CA Bar No. 297903)
(Admitted in this District)
jwaldrop@kasowitz.com
Darcy L. Jones (CA Bar No. 309474)
(Admitted in this District)
djones@kasowitz.com
Marcus A. Barber (CA Bar No. 307361)
(Admitted in this District)
mbarber@kasowitz.com
John W. Downing (CA Bar No. 252850)
(Admitted in this District)
jdowning@kasowitz.com
Heather S. Kim (CA Bar No. 277686)
(Admitted in this District)
hkim@kasowitz.com
Jack Shaw (CA Bar No. 309382)
(Admitted in this District)
jshaw@kasowitz.com
KASOWITZ BENSON TORRES LLP
333 Twin Dolphin Drive, Suite 200
Redwood Shores, California 94065
Telephone: (650) 453-5170
Facsimile: (650) 453-5171

Joshua A. Whitehill (NY Bar No. 4766473)
jwhitehill@kasowitz.com
(*Pro hac vice* admission)
Bradley P. Lerman (NY Bar No. 4906079)
blerman@kasowitz.com
(*Pro hac vice* admission)
Howard L. Bressler (NY Bar No. 248379)
hbressler@kasowitz.com
(*Pro hac vice* admission)
KASOWITZ BENSON TORRES LLP
1633 Broadway
New York, NY 10019
Telephone: (212) 506-1700
Facsimile: (212) 506-1800

Mark D. Siegmund (TX Bar No. 24117055)
mark@swclaw.com
STECKLER WAYNE COCHRAN CHERRY PLLC
8416 Old McGregor Road
Waco, TX 76712
Telephone: (254) 651-3690
Facsimile: (972) 387-4041

**Attorneys for Plaintiff
WSOU INVESTMENTS, LLC d/b/a
BRAZOS LICENSING AND
DEVELOPMENT**

CERTIFICATE OF SERVICE

A true and correct copy of the foregoing instrument was served or delivered electronically to all counsel of record, on this 26th day of October, 2021.

/s/ Jonathan K. Waldrop
Jonathan K. Waldrop

**Exhibit 1 to
WSOU Investments, LLC's
Amended Preliminary Infringement Contentions**

Infringement Claim Chart of U.S. Patent No. 8,712,708 (the “Asserted Patent”)

The Accused Instrumentalities include, without limitation, OnePlus Technology (Shenzhen) Co., Ltd. (“OnePlus” or “Defendant”), OnePlus Mobile Phones such as OnePlus 8, OnePlus 7 Pro, OnePlus 8 Pro, OnePlus 8T, OnePlus 9, OnePlus 9 Pro; all past, current and future OnePlus products and services that operate in the same or substantially similar manner as the specifically identified products and services; and all past, current and future OnePlus products and services that have the same or substantially similar features as the specifically identified products and services.

WSOU Investments, LLC (“WSOU” or “Plaintiff”) contends that OnePlus, including OnePlus’s employees, directly infringes each of the Asserted Claims, either literally or under the doctrine of equivalents. WSOU also contends that OnePlus has indirectly infringed and continues to indirectly infringe by contributing to and actively inducing infringement of one or more of the Asserted Claims.

WSOU does not intend this exemplary claim chart to be limiting, and WSOU reserves its rights to pursue other accused instrumentalities, patent claims, evidence, and infringement arguments in this case.

Exhibit(s)	Description	Link
Exhibit A	OnePlus 8 Specification Page	https://www.oneplus.com/8/specs
Exhibit B	Article on Warp Charge by Android Central	https://www.androidcentral.com/warp-charge
Exhibit C	Blog on Warp Charge 30 by Volta Charger	https://voltacharger.com/blogs/news/warp-charge-30-faster-charging-for-your-oneplus-device
Exhibit D	How to charge the battery correctly	https://forums.oneplus.com/threads/how-to-charge-the-battery-correctly.780695/
Exhibit E	GitHub OnePlus OSS Android Kernel Power Supply	https://github.com/OnePlusOSS/android_kernel_oneplus_sm8250/blob/cad09e061ef6cf689a4a1e54d27562e0f042236a/drivers/power/supply/power_supply_sysfs.c#L347
Exhibit F	GitHub OnePlus OSS Android Frameworks Base	https://github.com/OnePlusOSS/android_frameworks_base/blob/95ba353daa05fdd1c6011554ab7152fadfcdb932/services/core/java/com/android/server/am/BatteryStatsService.java#L337
Exhibit G	GitHub OnePlus OSS Android Kernel APM Power	https://github.com/OnePlusOSS/android_kernel_oneplus_sm8250/blob/cad09e061ef6cf689a4a1e54d27562e0f042236a/drivers/power/supply/apm_power.c
Exhibit H	GitHub OnePlus OSS Android Kernel Charger	https://github.com/OnePlusOSS/android_kernel_oneplus_sm8250/blob/d0cd6cd30f8e10b209461f24462ed316c76bc913/drivers/power/supply/axp288_charger.c#L363
Exhibit I	GitHub OnePlus OSS Android Frameworks Base Battery Statistics	https://github.com/OnePlusOSS/android_frameworks_base/blob/95ba353daa05fdd1c6011554ab7152fadfcdb932/core/java/com/android/internal/os/BatteryStatisticsImpl.java#L8705

Exhibit(s)	Description	Link
Exhibit J	Android Battery Statistics	https://android.googlesource.com/platform/frameworks/base.git/+/master/core/java/com/android/internal/os/BatteryStatsImpl.java
Exhibit K	OnePlus 9 Charging Test using Wrap Charger 65T	https://www.youtube.com/watch?v=CCHaMivzIvE
Exhibit L	Warp Charge 65T – Same Wattage but Better Performance	https://forums.oneplus.com/threads/deep-dive-lets-talk-charging.1403556/

Claims	OnePlus 8, including OnePlus Smartphone based on Android OS (The accused product)
1Pre. A method, comprising:	The accused product practices a method that comprises of detecting, by an apparatus, an availability of a charger adapter.
1a. detecting, by an apparatus, an availability of a charger adapter;	OnePlus provides smartphones such as OnePlus 8, OnePlus 7 Pro, OnePlus 8 Pro, OnePlus 8T, OnePlus 9, OnePlus 9 Pro that comes along with a OnePlus propriety charging adapter such as Warp Charge 30 Power Adapter. The Warp Charge 30 Power Adapter (i.e., charger adapter) delivers a 30-Watt (i.e., W) power output for fast charging the accused product. See Fig. 1 & Fig. 2. The accused product detects and displays the ‘Warp charging’ as it gets connect (i.e., available) with the Warp Charge 30 Power Adapter via the Warp type-c cable as shown in Fig. 3.

Citation 1: OnePlus 8

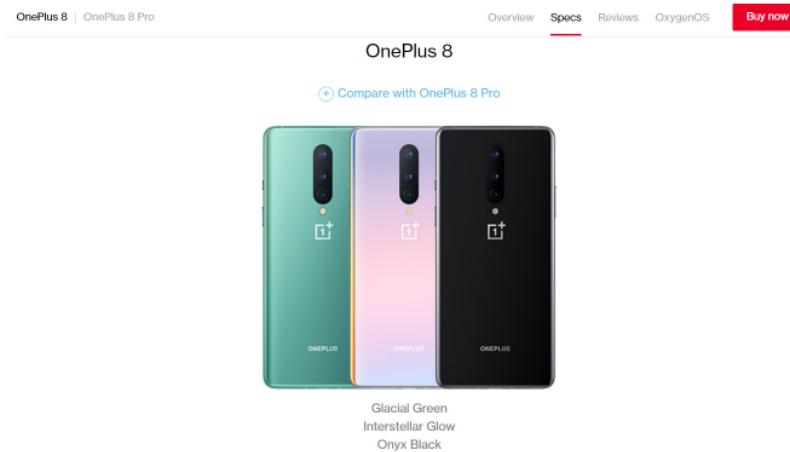


Fig. 1

Source: <https://www.oneplus.com/8/specs>, Page 1, Last Accessed May 18, 2021, Exhibit A

Citation 2: Warp Charge 30 Power Adapter

In The Box

OnePlus 8
Warp Charge 30 Power Adapter
Warp Type-C Cable (Support USB 2.0)
Quick Start Guide
Welcome Letter
Safety Information and Warranty Card
LOGO Sticker
Screen Protector
SIM Tray Ejector

Fig. 2

Source: <https://www.oneplus.com/8/specs>, Page 4, Last Accessed May 18, 2021, Exhibit A

Citation 3: Warp Charging display



Fig. 3

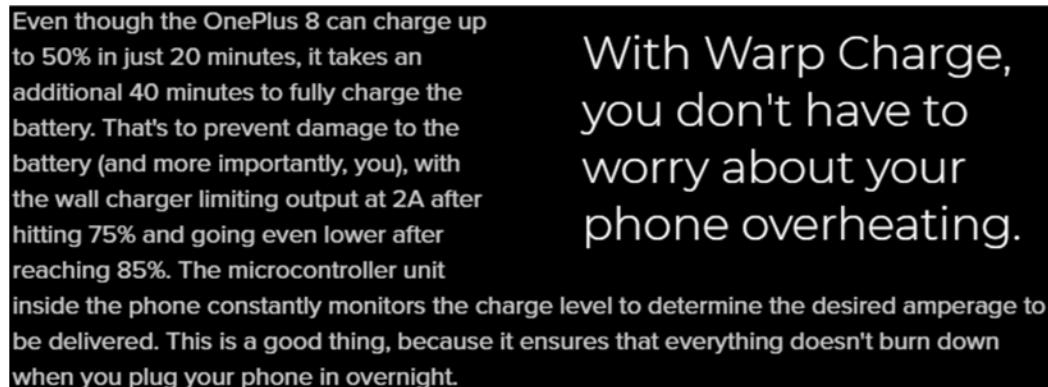
Source: <https://www.androidcentral.com/warp-charge>, Page 3, Last Accessed July 24, 2020, Exhibit B

1b. determining, by the apparatus, whether a battery charging point	The method that accused product practices comprises, whether a battery charging point is in a constant current phase or in a constant voltage phase, based on pre-determined battery charging characteristics, wherein the pre-
--	---

<p>is in a constant current phase or in a constant voltage phase, based on pre-determined battery charging characteristics, wherein the pre-determined battery charging characteristics comprise constant current phase charging characteristics and constant voltage phase charging characteristics;</p>	<p>determined battery charging characteristics comprise constant current phase charging characteristics and constant voltage phase charging characteristics.</p> <p>As described in the patent US'708B2 [Page 14, Line 18, Column 7] “<i>The capacity of a battery is the quantity of electrical charge stored in the battery at a given level. The capacity of a fully charged battery is usually measured in amp-hours (AH) or milliamp-hours (mAH) and is a measure of the size of the battery.</i>”</p> <p>The Warp Charge technology uses constant current charging (i.e., constant current phase) at multi-step of voltage open loop (or, multiple stages of voltage change). See Fig. 4.</p> <p>Citation 4: Warp Charge Uses Constant Current Charging</p> <p>Similar to Dash Charge, the Warp Charge technology is based on OPPO's Super Voltage Open Loop Multi-step Constant-Current Charging (Super VOOC) standard. At a regular voltage of 5V, Warp charge can attain 30W at 6A.</p> <p>Fig. 4</p> <p>Source: https://voltacharger.com/blogs/news/warp-charge-30-faster-charging-for-your-oneplus-device, Page 2, Last Accessed May 18, 2021, Exhibit C</p> <p>As described in the patent US'708B2 [Page 13, Line 14, Column 5] “<i>As the accumulated charge in the battery 160 increases, there comes a point at which the operation of the CC/CV charger IC 154 passes from the constant current (CC) phase to the constant voltage (CV) phase. In accordance with an embodiment of the invention, estimating the remaining charging time of the rechargeable battery 160 is based on determining whether the battery's charging point is in the constant current phase or in the constant voltage phase.</i>”</p>
---	---

OnePlus 8 includes a microcontroller unit that constantly monitor the charge level to determine the desired amperage (or, current) to be delivered. The accused product comprises a charging point (e.g. integrated circuit, or another hardware) that supplies constant current/constant voltage to the battery. The accused product switches to the constant voltage phase, when the accused product (or, microcontroller unit) have change in desired amperage to prevent battery from overheating and burn down. See Fig. 5.

Citation 5: Microcontroller unit determines desired amperage



Even though the OnePlus 8 can charge up to 50% in just 20 minutes, it takes an additional 40 minutes to fully charge the battery. That's to prevent damage to the battery (and more importantly, you), with the wall charger limiting output at 2A after hitting 75% and going even lower after reaching 85%. The microcontroller unit inside the phone constantly monitors the charge level to determine the desired amperage to be delivered. This is a good thing, because it ensures that everything doesn't burn down when you plug your phone in overnight.

With Warp Charge, you don't have to worry about your phone overheating.

Fig. 5

Source: <https://www.androidcentral.com/warp-charge>, Page 3, Last Accessed July 24, 2020, Exhibit B

Fig. 6 to Fig. 9 shows a snippet of OnePlus kernel source code that is installed onto the accused product. The accused product determines whether the power supply (or, Warp Charge 30 Power Adapter) is in constant current phase or constant voltage phase and associates the psp enum value accordingly.

Citation 6: psp enum value determination

```
17  
18 #define PSY_PROP(psy, prop, val) (power_supply_get_property(psy, \  
19 //POWER_SUPPLY_PROP_##prop, val))  
20  
21 #define _MPSY_PROP(prop, val) (power_supply_get_property(main_battery, \  
22 //prop, val))  
23  
24 #define MPSY_PROP(prop, val) _MPSY_PROP(POWER_SUPPLY_PROP_##prop, val)
```

Fig. 6

Source :

https://github.com/OnePlusOSS/android_kernel_oneplus_sm8250/blob/cad09e061ef6cf689a4a1e54d27562e0f042236a/drivers/power/supply/apm_power.c, Page 1, Last Accessed May 18, 2021, Exhibit G

Citation 7: Power Supply Attribute and Property

```
144 static ssize_t power_supply_show_property(struct device *dev,  
145 //                                     struct device_attribute *attr,  
146 //                                     char *buf) {  
147     ssize_t ret;  
148     struct power_supply *psy = dev_get_drvdata(dev);  
149     enum power_supply_property psp = attr - power_supply_attrs;  
150     union power_supply_propval value;
```

Fig. 7

Source:

https://github.com/OnePlusOSS/android_kernel_oneplus_sm8250/blob/cad09e061ef6cf689a4a1e54d27562e0f042236a/drivers/power/supply/power_supply_sysfs.c#L353, Page 2, Last Accessed May 18, 2021, Exhibit E

Citation 8: Device Attributes for Power Supply

```
298 /* Must be in the same order as POWER_SUPPLY_PROP_* */
299 static struct device_attribute power_supply_attrs[] = {
300     /* Properties of type 'int' */
301     POWER_SUPPLY_ATTR(status),
302     /* @bsp, 2018/07/13 Battery & Charging porting */
303     POWER_SUPPLY_ATTR(set_allow_read_extern_fg_iic),
304     POWER_SUPPLY_ATTR(cc_to_cv_point),
305     POWER_SUPPLY_ATTR(chg_protect_status),
```

Fig. 8

Source:

https://github.com/OnePlusOSS/android_kernel_oneplus_sm8250/blob/cad09e061ef6cf689a4a1e54d27562e0f042236a/drivers/power/supply/power_supply_sysfs.c#L353, Page 4, Last Accessed May 18, 2021, Exhibit E

Citation 9: psp enum value determines

```

363 static int axp288_charger_usb_set_property(struct power_supply *psy,
364                                         enum power_supply_property psp,
365                                         const union power_supply_propval *val)
366 {
367     struct axp288_chrg_info *info = power_supply_get_drvdata(psy);
368     int ret = 0;
369     int scaled_val;
370
371     switch (psp) {
372     case POWER_SUPPLY_PROP_CONSTANT_CHARGE_CURRENT:
373         scaled_val = min(val->intval, info->max_cc);
374         scaled_val = DIV_ROUND_CLOSEST(scaled_val, 1000);
375         ret = axp288_charger_set_cc(info, scaled_val);
376         if (ret < 0)
377             dev_warn(&info->pdev->dev, "set charge current failed\n");
378         break;
379     case POWER_SUPPLY_PROP_CONSTANT_CHARGE_VOLTAGE:
380         scaled_val = min(val->intval, info->max_cv);
381         scaled_val = DIV_ROUND_CLOSEST(scaled_val, 1000);
382         ret = axp288_charger_set_cv(info, scaled_val);
383         if (ret < 0)
384             dev_warn(&info->pdev->dev, "set charge voltage failed\n");
385         break;
386     case POWER_SUPPLY_PROP_INPUT_CURRENT_LIMIT:
387         ret = axp288_charger_set_vbus_inlmt(info, val->intval);
388         if (ret < 0)
389             dev_warn(&info->pdev->dev, "set input current limit failed\n");
390         break;
391     default:
392         ret = -EINVAL;
393     }

```

Fig. 9

Source:https://github.com/OnePlusOSS/android_kernel_oneplus_sm8250/blob/d0cd6cd30f8e10b209461f24462ed316c76bc913/drivers/power/supply/axp288_charger.c#L363, Page 5, Last Accessed May 18, 2021, Exhibit H

Fig. 10 and Fig. 11 shows snippets of OnePlus kernel source code that is installed onto the accused product. The accused product stores current and voltage attributes which are used to compute the constant current phase charging characteristics and constant voltage phase charging characteristics, respectively.

Citation 10: Constant Current Phase Charging Attributes

```

340     POWER_SUPPLY_ATTR(current_max),
341     POWER_SUPPLY_ATTR(current_now),
342     POWER_SUPPLY_ATTR(current_avg),
343     POWER_SUPPLY_ATTR(current_boot),

353     POWER_SUPPLY_ATTR(constant_charge_current),
354     POWER_SUPPLY_ATTR(constant_charge_current_max),

385     POWER_SUPPLY_ATTR(precharge_current),
386     POWER_SUPPLY_ATTR(charge_term_current),

404     POWER_SUPPLY_ATTR(input_current_max),
405     POWER_SUPPLY_ATTR(input_current_trim),
406     POWER_SUPPLY_ATTR(input_current_settled),

```

Fig. 10

Source:

https://github.com/OnePlusOSS/android_kernel_oneplus_sm8250/blob/cad09e061ef6cf689a4a1e54d27562e0f042236a/drivers/power/supply/power_supply_sysfs.c#L353, Page 5, Last Accessed May 18 2021, Exhibit E

Citation 11: Constant Voltage Phase Charging Attributes

```

332     POWER_SUPPLY_ATTR(voltage_max),
333     POWER_SUPPLY_ATTR(voltage_min),
334     POWER_SUPPLY_ATTR(voltage_max_design),
335     POWER_SUPPLY_ATTR(voltage_min_design),
336     POWER_SUPPLY_ATTR(voltage_now),
337     POWER_SUPPLY_ATTR(voltage_avg),
338     POWER_SUPPLY_ATTR(voltage_ocv),
339     POWER_SUPPLY_ATTR(voltage_boot),
355     POWER_SUPPLY_ATTR(constant_charge_voltage),
356     POWER_SUPPLY_ATTR(constant_charge_voltage_max),
503     POWER_SUPPLY_ATTR(voltage_max_limit),

```

Fig. 11

Source:

https://github.com/OnePlusOSS/android_kernel_oneplus_sm8250/blob/cad09e061ef6cf689a4a1e54d27562e0f042236a/drivers/power/power_supply/power_supply_sysfs.c#L353, Page 4, 5 & 6, Last Accessed May 18, 2021,

Exhibit E

The accused product uses various characteristics such as battery percentage upon the time of charging, the time period over which the battery is charging, battery voltage to determine that if the charging point is in the constant current phase or the constant voltage phase. See Fig. 12 and Fig. 13.

Citation 12: Battery percentage on OnePlus 8



Fig. 12

Source: <https://www.androidcentral.com/warp-charge>, Page 3, Last Accessed July 24, 2020, Exhibit B

	<p>Citation 13: Constant Current and Constant Voltage Phase</p> <p>4) The process of charging li-ion batt goes like this (let's start from full current):</p> <p>a) CC (constant current) stage which means the battery takes full current, in this case 4A but frankly it's 3,4-3,6A. For 3,3Ah battery it's current around 1,1C, it's perfectly good value for battery life.</p> <p>b) when battery reaches certain point of voltage (on Dash I noticed it's around 60-70% approx.), current drops naturally and it keeps going down with increasing voltage</p> <p>c) CV (constant voltage) means BMS sets 4,2V (usual value for 100% capacity of lithium cell) and current is going to 0A since your battery is almost charged at this point e.g 4,1V and charger gives 4,2V which is $4,2-4,1V = 0,1V$, it allows really low current. But it's important to charge to 100% because last stage lasts long but it gives you significantly more capacity, especially for only 3,3Ah cell. Moreover, this stage is most dangerous because voltage for almost fully charged cell becomes unstable and when it takes current, it'll increase voltage easily and rapidly. That's why cut off current has to work and stops charging when it's time. The process can go a few times e.g. if battery goes $>4,2V$, the charger disconnects, waits few seconds, lower current from charger and starts charging again. And after e.g. 3 repeats, it ends the charging process for good.</p>
1c. calculating, by the apparatus, a time remaining to charge in the constant current phase based on the constant current phase characteristics, if the accused product practices comprises, calculating a time remaining to charge in the constant current phase based on the constant current phase charging characteristics, if the battery charging point is in the constant current phase, and calculating a time remaining to charge in the constant voltage phase based on the constant voltage phase charging characteristics, if the battery charging point is in the constant voltage phase.	<p>Fig. 13</p> <p>Source: https://forums.oneplus.com/threads/how-to-charge-the-battery-correctly.780695/, Page 3, Last Accessed May 18, 2021, Exhibit D</p>

<p>battery charging point is in the constant current phase; and</p> <p>1d. calculating, by the apparatus, a time remaining to charge in the constant voltage phase based on the constant voltage phase charging characteristics, if the battery charging point is in the constant voltage phase;</p>	<p>the smartphone, and more to estimate the time remaining to charge the smartphone's battery in the near real-time. See Fig. 14 & Fig. 15.</p> <p style="text-align: center;">Citation 14: Warp Charging Displays on OnePlus 8</p>  <p style="text-align: center;">Fig. 14</p> <p>Source: https://www.androidcentral.com/warp-charge, Page 3, Last Accessed July 24, 2020, Exhibit B</p>
---	--

	<p>Citation 15: Non-removable 4300 mAh battery capacity</p> <table> <thead> <tr> <th>Performance</th><th>Operating System: OxygenOS based on Android™ 10 CPU: Qualcomm® Snapdragon™865 5G Chipset: X55 GPU: Adreno 650 RAM: 8GB/12GB LPDDR4X Storage: 128GB/256GB UFS 3.0 2-LANE Battery: 4300 mAh (non-removable)</th></tr> </thead> </table>	Performance	Operating System: OxygenOS based on Android™ 10 CPU: Qualcomm® Snapdragon™865 5G Chipset: X55 GPU: Adreno 650 RAM: 8GB/12GB LPDDR4X Storage: 128GB/256GB UFS 3.0 2-LANE Battery: 4300 mAh (non-removable)
Performance	Operating System: OxygenOS based on Android™ 10 CPU: Qualcomm® Snapdragon™865 5G Chipset: X55 GPU: Adreno 650 RAM: 8GB/12GB LPDDR4X Storage: 128GB/256GB UFS 3.0 2-LANE Battery: 4300 mAh (non-removable)		
	<p>Fig. 15</p> <p>Source: https://www.oneplus.com/8/specs, Page 1-2, Last Accessed May 18, 2021, Exhibit A</p> <p>As described in the patent US'708B2 [Page 15, Line 57, Column 10] “<i>Step 408: if the battery charging point is in the constant current phase, calculating a time remaining to charge in the constant current phase based on the constant current phase charging characteristics and a time remaining to charge in the constant Voltage phase based on the constant Voltage phase charging characteristics. In step 408 the present capacity and the present charging current are also inputs for this calculation.</i>” and [Page 15, Line 65, Column 10] “<i>Step 410: if the battery charging point is in the constant Voltage phase, calculating a time remaining to charge in the constant Voltage phase based on the constant Voltage phase charging characteristics. In step 410 the present charging current is also an input for this calculation.</i>”</p> <p>The accused product (e.g., OnePlus 3T) calculates the time to charge the device to 100% battery level. See Fig. 16.</p> <p>Citation 16: Estimation of time left to reach the 100% battery level</p>		

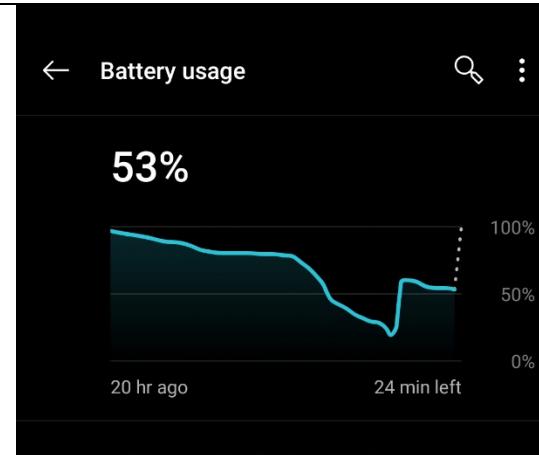


Fig. 16

Source: Snippet taken from OnePlus 3T

The Fig. 17, shows the estimate of remaining charging time on lock screen of the accused product (e.g, OnePlus 9).

Citation 17: OnePlus 9 lock screen display remaining charging time estimate

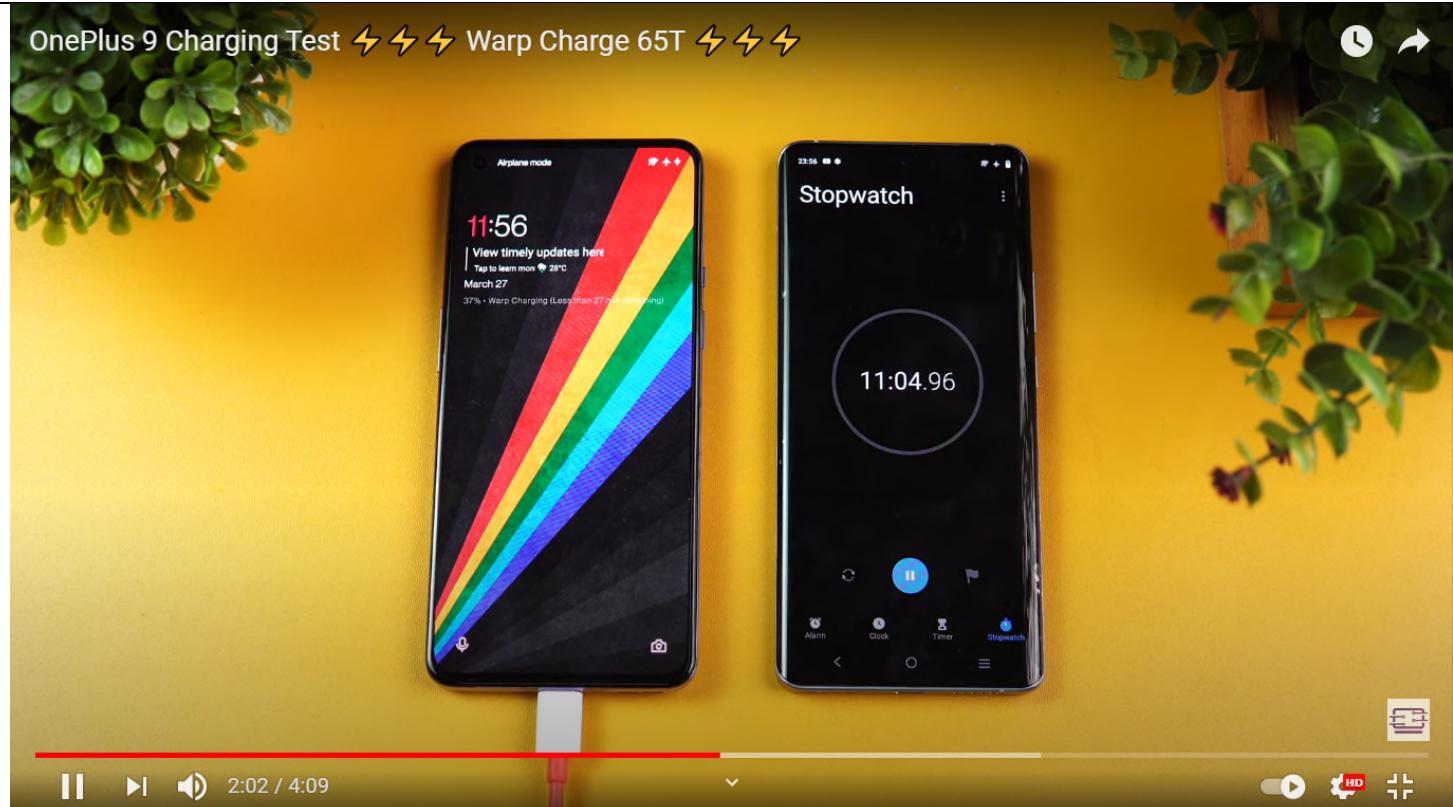


Fig. 17

Source: <https://www.youtube.com/watch?v=CCHaMivzIvE>, Time- 2:02, Last accessed June 24, 2021, Exhibit K

The accused product determines various characteristics (i.e. constant voltage phase charging characteristics) such as voltage that the charging point feeds, battery level, total battery capacity, and more such parameters at the constant current phase. Moreover, the accused product also determines various characteristics (i.e. constant

current phase charging characteristics) such as current that the charging point feeds, battery level, total battery capacity, and more at the constant voltage phase. See Fig. 18 to **Error! Reference source not found..**

Citation 18: Constant Current Phase Charging Attributes

```

340     POWER_SUPPLY_ATTR(current_max),
341     POWER_SUPPLY_ATTR(current_now),
342     POWER_SUPPLY_ATTR(current_avg),
343     POWER_SUPPLY_ATTR(current_boot),
353     POWER_SUPPLY_ATTR(constant_charge_current),
354     POWER_SUPPLY_ATTR(constant_charge_current_max),
385     POWER_SUPPLY_ATTR(precharge_current),
386     POWER_SUPPLY_ATTR(charge_term_current),
404     POWER_SUPPLY_ATTR(input_current_max),
405     POWER_SUPPLY_ATTR(input_current_trim),
406     POWER_SUPPLY_ATTR(input_current_settled),

```

Fig. 18

Source:

https://github.com/OnePlusOSS/android_kernel_oneplus_sm8250/blob/cad09e061ef6cf689a4a1e54d27562e0f042236a/drivers/power/supply/power_supply_sysfs.c#L353, Page 5, Last Accessed May 18, 2021, Exhibit E

Citation 19: Constant Voltage Phase Charging Attributes

```
332     POWER_SUPPLY_ATTR(voltage_max),  
333     POWER_SUPPLY_ATTR(voltage_min),  
334     POWER_SUPPLY_ATTR(voltage_max_design),  
335     POWER_SUPPLY_ATTR(voltage_min_design),  
336     POWER_SUPPLY_ATTR(voltage_now),  
337     POWER_SUPPLY_ATTR(voltage_avg),  
338     POWER_SUPPLY_ATTR(voltage_ocv),  
339     POWER_SUPPLY_ATTR(voltage_boot),  
  
355     POWER_SUPPLY_ATTR(constant_charge_voltage),  
356     POWER_SUPPLY_ATTR(constant_charge_voltage_max),  
  
503     POWER_SUPPLY_ATTR(voltage_max_limit),
```

Fig. 19

Source:

https://github.com/OnePlusOSS/android_kernel_oneplus_sm8250/blob/cad09e061ef6cf689a4a1e54d27562e0f042236a/drivers/power/power_supply/power_supply_sysfs.c#L353, Page 4, 5 & 6, Last Accessed May 18, 2021,

Exhibit E

Citation 20: Stored Charge and Current Charge

	<pre> 346 POWER_SUPPLY_ATTR(charge_full_design), 347 POWER_SUPPLY_ATTR(charge_empty_design), 348 POWER_SUPPLY_ATTR(charge_full), 349 POWER_SUPPLY_ATTR(charge_empty), 350 POWER_SUPPLY_ATTR(charge_now), 351 POWER_SUPPLY_ATTR(charge_avg), 352 POWER_SUPPLY_ATTR(charge_counter), 353 POWER_SUPPLY_ATTR(constant_charge_current), 354 POWER_SUPPLY_ATTR(constant_charge_current_max), 355 POWER_SUPPLY_ATTR(constant_charge_voltage), 356 POWER_SUPPLY_ATTR(constant_charge_voltage_max), 357 POWER_SUPPLY_ATTR(charge_control_limit), 358 POWER_SUPPLY_ATTR(charge_control_limit_max), </pre>
--	---

Fig. 20

Source:

https://github.com/OnePlusOSS/android_kernel_oneplus_sm8250/blob/cad09e061ef6cf689a4a1e54d27562e0f042236a/drivers/power/supply/power_supply_sysfs.c#L353, Page 5, Last Accessed May 18, 2021, Exhibit E

As an example, shown in Fig. 21 is a code snippet that shows the calculation of charging time estimate. Upon information and belief, the accused product uses ‘mChargeStepTracker’ to estimate (or, calculate) the time remaining to charge the battery, based on constant current/voltage characteristics. See Fig. 22.

Citation 21: Battery Charging and Discharging Attributes

```

8705     public long computeChargeTimeRemaining(long curTime) {
8706         if (mOnBattery) {
8707             // Not yet working.
8708             return -1;
8709         }
8710         /* Broken
8711         int curLevel = mCurrentBatteryLevel;
8712         int plugLevel = mDischargePlugLevel;
8713         if (plugLevel < 0 || curLevel < (plugLevel+1)) {
8714             return -1;
8715         }
8716         long duration = computeBatteryRealtime(curTime, STATS_SINCE_UNPLUGGED);
8717         if (duration < 1000*1000) {
8718             return -1;
8719         }
8720         long usPerLevel = duration/(curLevel-plugLevel);
8721         return usPerLevel * (100-curLevel);
8722         */
8723         if (mChargeStepTracker.mNumStepDurations < 1) {
8724             return -1;
8725         }
8726         long msPerLevel = mChargeStepTracker.computeTimePerLevel();
8727         if (msPerLevel <= 0) {
8728             return -1;
8729         }
8730         return (msPerLevel * (100-mCurrentBatteryLevel)) * 1000;
8731     }

```

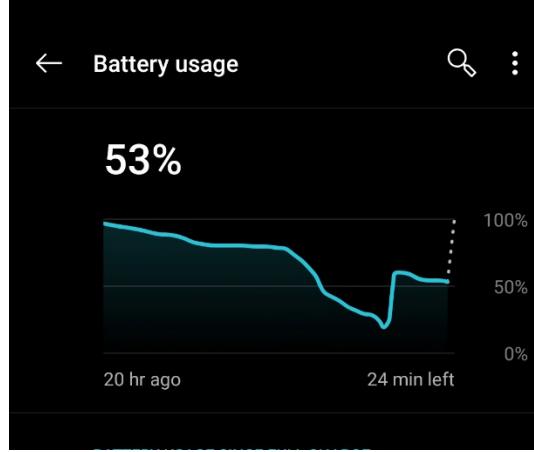
Fig. 21

Source:

https://github.com/OnePlusOSS/android_frameworks_base/blob/95ba353daa05fdd1c6011554ab7152fadfcdb932/core/java/com/android/internal/os/BatteryStatsImpl.java#L8705, Page 243-244, Last Accessed May 18, 2021,

Exhibit I

	<p>Citation 22: Constant Current Phase and Constant Voltage Phase</p> <p>4) The process of charging li-ion batt goes like this (let's start from full current):</p> <ul style="list-style-type: none"> a) CC (constant current) stage which means the battery takes full current, in this case 4A but frankly it's 3,4-3,6A. For 3,3Ah battery it's current around 1,1C, it's perfectly good value for battery life. b) when battery reaches certain point of voltage (on Dash I noticed it's around 60-70% approx.), current drops naturally and it keeps going down with increasing voltage c) CV (constant voltage) means BMS sets 4,2V (usual value for 100% capacity of lithium cell) and current is going to 0A since your battery is almost charged at this point e.g 4,1V and charger gives 4,2V which is $4,2-4,1V = 0,1V$, it allows really low current. But it's important to charge to 100% because last stage lasts long but it gives you significantly more capacity, especially for only 3,3Ah cell. Moreover, this stage is most dangerous because voltage for almost fully charged cell becomes unstable and when it takes current, it'll increase voltage easily and rapidly. That's why cut off current has to work and stops charging when it's time. The process can go a few times e.g. if battery goes $>4,2V$, the charger disconnects, waits few seconds, lower current from charger and starts charging again. And after e.g. 3 repeats, it ends the charging process for good. <p style="text-align: center;">Fig. 22</p> <p>Source: https://forums.oneplus.com/threads/how-to-charge-the-battery-correctly.780695/, Page 3, Last Accessed May 18, 2021, Exhibit D</p>
<p>1e. wherein the time remaining to charge in the constant current phase is based on stored charge characteristics in the constant current phase, when the battery</p>	<p>The accused product practices a method, wherein the time remaining to charge in the constant current phase is based on stored charge characteristics in the constant current phase, when the battery charging point is in the constant current phase and wherein battery stored charge characteristics comprises a battery stored charge value based on monitored tracking of battery charging and discharging.</p> <p>As described in the patent US'708B2 [Page 14, Line 48, Column 7] “<i>Along with the measured data for datasets 210 and 215, the following information may be stored in the device 100 as dataset 220 for each combination of</i></p>

<p>charging point is in the constant current phase and wherein battery stored charge characteristics comprises a battery stored charge value based on monitored tracking of battery charging and discharging.</p>	<p><i>device 100 battery 160 type: 1 Charge current during CC phase (IBATCC); 2) Battery 160 voltage when the charging switches to the CV phase (VBATCV). If the battery 160 voltage slightly increases during the CV phase, a value of the lowest Voltage in the constant Voltage phase may be chosen; and 3Duration of the CV phase (TCVTOT)."</i></p> <p>The accused product calculates and displays the amount of time life to charge the battery to 100% charge level as shown in See Fig. 23</p> <p>Citation 23: Estimate time left to reach the 100% battery level</p>  <p>Fig. 23</p> <p>Source: Snippet taken from OnePlus 3T</p> <p>OnePlus 8 is deployed with Android based Operating System (i.e., OS). See Fig. 24.</p>
---	--

Citation 24: OnePlus 8 runs Android based OS

Performance

Operating System: OxygenOS based on Android™ 10
CPU: Qualcomm® Snapdragon™865
5G Chipset: X55
GPU: Adreno 650
RAM: 8GB/12GB LPDDR4X
Storage: 128GB/256GB UFS 3.0 2-LANE
Battery: 4300 mAh (non-removable)

Fig. 24

Source: <https://www.oneplus.com/8/specs>, Page 1-2, Last Accessed May 18, 2021, Exhibit A

Fig. 25 and Fig. 26 show snippets of OnePlus kernel source code that is installed onto the accused product, and the accused product stores battery stored charge value, which is based on the attributes like “mCurrentBatteryLevel” and “mDischargePlugLevel” (i.e., battery charging and discharging).

Citation 25: Stored Charge and Current Charge

	<pre>346 POWER_SUPPLY_ATTR(charge_full_design), 347 POWER_SUPPLY_ATTR(charge_empty_design), 348 POWER_SUPPLY_ATTR(charge_full), 349 POWER_SUPPLY_ATTR(charge_empty), 350 POWER_SUPPLY_ATTR(charge_now), 351 POWER_SUPPLY_ATTR(charge_avg), 352 POWER_SUPPLY_ATTR(charge_counter), 353 POWER_SUPPLY_ATTR(constant_charge_current), 354 POWER_SUPPLY_ATTR(constant_charge_current_max), 355 POWER_SUPPLY_ATTR(constant_charge_voltage), 356 POWER_SUPPLY_ATTR(constant_charge_voltage_max), 357 POWER_SUPPLY_ATTR(charge_control_limit), 358 POWER_SUPPLY_ATTR(charge_control_limit_max),</pre>
--	---

Fig. 25

Source:

https://github.com/OnePlusOSS/android_kernel_oneplus_sm8250/blob/cad09e061ef6cf689a4a1e54d27562e0f042236a/drivers/power/supply/power_supply_sysfs.c#L353, Page 5, Last Accessed May 18, 2021, Exhibit E

Citation 26: Battery Charging and Discharging Attributes

```

8705     public long computeChargeTimeRemaining(long curTime) {
8706         if (mOnBattery) {
8707             // Not yet working.
8708             return -1;
8709         }
8710         /* Broken
8711         int curLevel = mCurrentBatteryLevel;
8712         int plugLevel = mDischargePlugLevel;
8713         if (plugLevel < 0 || curLevel < (plugLevel+1)) {
8714             return -1;
8715         }
8716         long duration = computeBatteryRealtime(curTime, STATS_SINCE_UNPLUGGED);
8717         if (duration < 1000*1000) {
8718             return -1;
8719         }
8720         long usPerLevel = duration/(curLevel-plugLevel);
8721         return usPerLevel * (100-curLevel);
8722         */
8723         if (mChargeStepTracker.mNumStepDurations < 1) {
8724             return -1;
8725         }
8726         long msPerLevel = mChargeStepTracker.computeTimePerLevel();
8727         if (msPerLevel <= 0) {
8728             return -1;
8729         }
8730         return (msPerLevel * (100-mCurrentBatteryLevel)) * 1000;
8731     }

```

Fig. 26

Source:

https://github.com/OnePlusOSS/android_frameworks_base/blob/95ba353daa05fdd1c6011554ab7152fadfcdb932/core/java/com/android/internal/os/BatteryStatsImpl.java#L8705, Page 243-244, Last Accessed May 18, 2021,

Exhibit I

Fig. 27 shows the snippet of source code of Android that computes and stores the time in millisecond (i.e. ms) requires for charging the battery by 1%, in ‘msPerLevel’, which depends on both charging and discharging.

As an example, the time to charge 1% depends on the charging of the battery, as with increased power supply there is a decrease in the displayed charging time. Also, the time to charge 1% depends on the discharging of the battery caused by the processing load, resulting in an increase in the battery's charging time. The accused product running android-based OS computes the time remaining to charge battery, in constant current, by multiplying the ‘msPerLevel’ with battery level remaining to charge. See Fig. 27 - Fig. 30.

Citation 27: Computation of remaining time based on per step level

```

12572     @Override
12573     public long computeChargeTimeRemaining(long curTime) {
12574         if (mOnBattery) {
12575             // Not yet working.
12576             return -1;
12577         }
12578         if (mBatteryTimeToFullSeconds >= 0) {
12579             return mBatteryTimeToFullSeconds * (1000 * 1000); // s to us
12580         }
12581         // Else use algorithmic approach
12582         if (mChargeStepTracker.mNumStepDurations < 1) {
12583             return -1;
12584         }
12585         long msPerLevel = mChargeStepTracker.computeTimePerLevel();
12586         if (msPerLevel <= 0) {
12587             return -1;
12588         }
12589         return (msPerLevel * (100 - mCurrentBatteryLevel)) * 1000;
12590     }

```

Fig. 27

Source:

<https://android.googlesource.com/platform/frameworks/base.git/+/master/core/java/com/android/internal/os/BatteryStatsImpl.java#12604>, Page 1, Last Accessed May 18, 2021, Exhibit J

Citation 28: Battery Charging and Discharging Attributes

```

8705     public long computeChargeTimeRemaining(long curTime) {
8706         if (mOnBattery) {
8707             // Not yet working.
8708             return -1;
8709         }
8710         /* Broken
8711         int curLevel = mCurrentBatteryLevel;
8712         int plugLevel = mDischargePlugLevel;
8713         if (plugLevel < 0 || curLevel < (plugLevel+1)) {
8714             return -1;
8715         }
8716         long duration = computeBatteryRealtime(curTime, STATS_SINCE_UNPLUGGED);
8717         if (duration < 1000*1000) {
8718             return -1;
8719         }
8720         long usPerLevel = duration/(curLevel-plugLevel);
8721         return usPerLevel * (100-curLevel);
8722         */
8723         if (mChargeStepTracker.mNumStepDurations < 1) {
8724             return -1;
8725         }
8726         long msPerLevel = mChargeStepTracker.computeTimePerLevel();
8727         if (msPerLevel <= 0) {
8728             return -1;
8729         }
8730         return (msPerLevel * (100-mCurrentBatteryLevel)) * 1000;
8731     }

```

Fig. 28

Source:

https://github.com/OnePlusOSS/android_frameworks_base/blob/95ba353daa05fdd1c6011554ab7152fadfcdb932/core/java/com/android/internal/os/BatteryStatsImpl.java#L8705, Page 243-244, Last Accessed May 18, 2021

Exhibit I

Citation 29: BatteryStats.Java uses BatteryStatsImpl data to provide

```
76     static IBatteryStats sService;  
77     final BatteryStatsImpl mStats;  
78     final BatteryStatsHandler mHandler;  
79     Context mContext;  
80     PowerManagerInternal mPowerManagerInternal;
```

Fig. 29

Source:

https://github.com/OnePlusOSS/android_frameworks_base/blob/95ba353daa05fdd1c6011554ab7152fadfcdb932/services/core/java/com/android/server/am/BatteryStatsService.java#L337, Page 3, Last Accessed May 18, 2021, Exhibit F

Citation 30: BatteryStats.Java uses BatteryStatsImpl data to provide

```
337     public long computeChargeTimeRemaining() {  
338         synchronized (mStats) {  
339             long time = mStats.computeChargeTimeRemaining(SystemClock.elapsedRealtime());  
340             return time >= 0 ? (time/1000) : time;  
341         }  
342     }
```

Fig. 30

Source:

https://github.com/OnePlusOSS/android_frameworks_base/blob/95ba353daa05fdd1c6011554ab7152fadfcdb93

	<p>2/services/core/java/com/android/server/am/BatteryStatsService.java#L337, Page 9, Last Accessed May 18, 2021, Exhibit F</p>
2. The method of claim 1, wherein the battery stored charge characteristics comprise data that provides a remaining charging time estimate based on a battery stored charge estimate.	<p>The accused product practices a method, wherein the battery stored charge characteristics comprise data that provides a remaining charging time estimate based on a battery stored charge estimate.</p> <p>Fig. 31 shows snippets of the OnePlus kernel source code that is installed onto the accused product, and the accused product stores battery stored charge value and accordingly “computeChargeTimeRemaining” (i.e., remaining charging time) is estimated. See Fig. 32.</p> <p style="text-align: center;">Citation 31: Stored Charge and Current Charge</p> <pre> 346 POWER_SUPPLY_ATTR(charge_full_design), 347 POWER_SUPPLY_ATTR(charge_empty_design), 348 POWER_SUPPLY_ATTR(charge_full), 349 POWER_SUPPLY_ATTR(charge_empty), 350 POWER_SUPPLY_ATTR(charge_now), 351 POWER_SUPPLY_ATTR(charge_avg), 352 POWER_SUPPLY_ATTR(charge_counter), 353 POWER_SUPPLY_ATTR(constant_charge_current), 354 POWER_SUPPLY_ATTR(constant_charge_current_max), 355 POWER_SUPPLY_ATTR(constant_charge_voltage), 356 POWER_SUPPLY_ATTR(constant_charge_voltage_max), 357 POWER_SUPPLY_ATTR(charge_control_limit), 358 POWER_SUPPLY_ATTR(charge_control_limit_max), </pre> <p style="text-align: center;">Fig. 31</p>

Source:

https://github.com/OnePlusOSS/android_kernel_oneplus_sm8250/blob/cad09e061ef6cf689a4a1e54d27562e0f042236a/drivers/power/supply/power_supply_sysfs.c#L353, Page 5, Last Accessed May 18, 2021, Exhibit E

Citation 32: Remaining Charging Time Estimation

```

8705     public long computeChargeTimeRemaining(long curTime) {
8706         if (mOnBattery) {
8707             // Not yet working.
8708             return -1;
8709         }
8710         /* Broken
8711         int curLevel = mCurrentBatteryLevel;
8712         int plugLevel = mDischargePlugLevel;
8713         if (plugLevel < 0 || curLevel < (plugLevel+1)) {
8714             return -1;
8715         }
8716         long duration = computeBatteryRealtime(curTime, STATS_SINCE_UNPLUGGED);
8717         if (duration < 1000*1000) {
8718             return -1;
8719         }
8720         long usPerLevel = duration/(curLevel-plugLevel);
8721         return usPerLevel * (100-curLevel);
8722         */
8723         if (mChargeStepTracker.mNumStepDurations < 1) {
8724             return -1;
8725         }
8726         long msPerLevel = mChargeStepTracker.computeTimePerLevel();
8727         if (msPerLevel <= 0) {
8728             return -1;
8729         }
8730         return (msPerLevel * (100-mCurrentBatteryLevel)) * 1000;
8731     }

```

Fig. 32

	<p>Source:</p> <p>https://github.com/OnePlusOSS/android_frameworks_base/blob/95ba353daa05fdd1c6011554ab7152fadfcdb932/core/java/com/android/internal/os/BatteryStatsImpl.java#L8705, Page 243-244, Last Accessed May 18, 2021,</p> <p style="text-align: center;">Exhibit I</p>
<p>4. The method of claim 1, wherein the time remaining to charge in the constant voltage phase is based on charging current characteristics in the constant voltage phase, when the battery charging point is in the constant voltage phase.</p>	<p>The accused product practices a method, wherein the time remaining to charge in the constant voltage phase is based on charging current characteristics in the constant voltage phase, when the battery charging point is in the constant voltage phase.</p> <p>Fig. 33 to Fig. 36 shows snippets of OnePlus kernel source code that is installed onto the accused product. The accused product determines whether the power supply (or, Warp Charge 30 Power Adapter) is in constant current phase or constant voltage phase by using ‘power_supply_property psp’ which comprises attributes for the battery charging characteristics and associates the psp enum value accordingly.</p> <p style="text-align: center;">Citation 33: psp enum value determination</p> <pre> 17 18 #define PSY_PROP(psy, prop, val) (power_supply_get_property(psy, \ 19 POWER_SUPPLY_PROP_##prop, val)) 20 21 #define _MPSY_PROP(prop, val) (power_supply_get_property(main_battery, \ 22 prop, val)) 23 24 #define MPSY_PROP(prop, val) _MPSY_PROP(POWER_SUPPLY_PROP_##prop, val) </pre> <p style="text-align: center;">Fig. 33</p>

Source:

https://github.com/OnePlusOSS/android_kernel_oneplus_sm8250/blob/cad09e061ef6cf689a4a1e54d27562e0f042236a/drivers/power/supply/apm_power.c, Page 1, Last Accessed May 18, 2021, Exhibit G

Citation 34: Power Supply Property

```
144 static ssize_t power_supply_show_property(struct device *dev,  
145                                         struct device_attribute *attr,  
146                                         char *buf) {  
147     ssize_t ret;  
148     struct power_supply *psy = dev_get_drvdata(dev);  
149     enum power_supply_property psp = attr->power_SUPPLY_ATTRS;  
150     union power_supply_propval value;
```

Fig. 34

Source: https://github.com/OnePlusOSS/android_kernel_oneplus_sm8250/blob/cad09e061ef6cf689a4a1e54d27562e0f042236a/drivers/power/supply/power_supply_sysfs.c#L353, Page 2, Last Accessed May 18, 2021,

Exhibit E

Citation 35: Device Attributes for Power Supply

```
298     /* Must be in the same order as POWER_SUPPLY_PROP_* */
299     static struct device_attribute power_supply_attrs[] = {
300         /* Properties of type 'int' */
301         POWER_SUPPLY_ATTR(status),
302         /* @bsp, 2018/07/13 Battery & Charging porting */
303         POWER_SUPPLY_ATTR(set_allow_read_extern_fg_iic),
304         POWER_SUPPLY_ATTR(cc_to_cv_point),
305         POWER_SUPPLY_ATTR(chg_protect_status),
```

Fig. 35

Source:

https://github.com/OnePlusOSS/android_kernel_oneplus_sm8250/blob/cad09e061ef6cf689a4a1e54d27562e0f042236a/drivers/power/power_supply/power_supply_sysfs.c#L353, Page 4, Last Accessed May 18, 2021, Exhibit E

Citation 36: Constant Current & Constant Voltage in Power Supply Property

```

*** 363 static int axp288_charger_usb_set_property(struct power_supply *psy,
364                                     enum power_supply_property psp,
365                                     const union power_supply_propval *val)
366 {
367     struct axp288_chrg_info *info = power_supply_get_drvdata(psy);
368     int ret = 0;
369     int scaled_val;
370
371     switch (psp) {
372     case POWER_SUPPLY_PROP_CONSTANT_CHARGE_CURRENT:
373         scaled_val = min(val->intval, info->max_cc);
374         scaled_val = DIV_ROUND_CLOSEST(scaled_val, 1000);
375         ret = axp288_charger_set_cc(info, scaled_val);
376         if (ret < 0)
377             dev_warn(&info->pdev->dev, "set charge current failed\n");
378         break;
379     case POWER_SUPPLY_PROP_CONSTANT_CHARGE_VOLTAGE:
380         scaled_val = min(val->intval, info->max_cv);
381         scaled_val = DIV_ROUND_CLOSEST(scaled_val, 1000);
382         ret = axp288_charger_set_cv(info, scaled_val);
383         if (ret < 0)
384             dev_warn(&info->pdev->dev, "set charge voltage failed\n");
385         break;
386     case POWER_SUPPLY_PROP_INPUT_CURRENT_LIMIT:
387         ret = axp288_charger_set_vbus_inlmt(info, val->intval);
388         if (ret < 0)
389             dev_warn(&info->pdev->dev, "set input current limit failed\n");
390         break;
391     default:
392         ret = -EINVAL;
393     }

```

Fig. 36

Source:https://github.com/OnePlusOSS/android_kernel_oneplus_sm8250/blob/d0cd6cd30f8e10b209461f24462ed316c76bc913/drivers/power/supply/axp288_charger.c#L363, Page 5, Last Accessed May 18, 2021, Exhibit H

When the accused product is connected to the charger, firstly it takes full current, but when the battery reaches a certain point of voltage, the type of charging switches to constant voltage from constant current. In the CV (i.e., constant voltage) mode, the voltage is set to 4.2V, and accordingly current (i.e., charging current characteristics) is allowed to charge device to 100%. See Fig. 37.

Citation 37: Constant Current Phase and Constant Voltage Phase

- 4) The process of charging li-ion batt goes like this (let's start from full current):
- a) CC (constant current) stage which means the battery takes full current, in this case 4A but frankly it's 3,4-3,6A.
For 3,3Ah battery it's current around 1,1C, it's perfectly good value for battery life.
 - b) when battery reaches certain point of voltage (on Dash I noticed it's around 60-70% approx.), current drops naturally and it keeps going down with increasing voltage
 - c) CV (constant voltage) means BMS sets 4,2V (usual value for 100% capacity of lithium cell) and current is going to 0A since your battery is almost charged at this point e.g 4,1V and charger gives 4,2V which is $4,2-4,1V = 0,1V$, it allows really low current. But it's important to charge to 100% because last stage lasts long but it gives you significantly more capacity, especially for only 3,3Ah cell. Moreover, this stage is most dangerous because voltage for almost fully charged cell becomes unstable and when it takes current, it'll increase voltage easily and rapidly. That's why cut off current has to work and stops charging when it's time. The process can go a few times e.g. if battery goes $>4,2V$, the charger disconnects, waits few seconds, lower current from charger and starts charging again. And after e.g. 3 repeats, it ends the charging process for good.

Fig. 37

Source: <https://forums.oneplus.com/threads/how-to-charge-the-battery-correctly.780695/>, Page 3, Last Accessed May 18, 2021, Exhibit D

	<p>Citation 38: Estimation of time left to reach the 100% battery level</p>
<p>5Pre. The method of claim 1, further comprising:</p> <p>5a. identifying, by the apparatus, a correct category of the charger adapter after detecting its availability; and</p>	<p>The accused product practices a method comprising identifying, by the apparatus, a correct category of the charger adapter after detecting its availability.</p> <p>OnePlus provides smartphones such as OnePlus 8 that come along with a Warp Charge 30 Power Adapter. The Warp Charge 30 Power Adapter (i.e., charger adapter) delivers a 30-Watt power output for fast charging the accused product. See Fig. 39 and Fig. 40.</p>

Citation 39: OnePlus 8

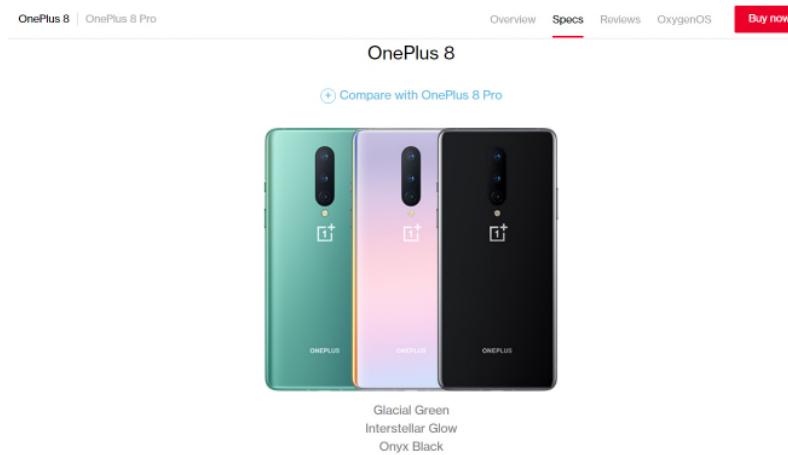


Fig. 39

Source: <https://www.oneplus.com/8/specs>, Page 1, Last Accessed May 18, 2021, Exhibit A

Citation 40: Warp Charge 30 Power Adapter

In The Box

OnePlus 8
Warp Charge 30 Power Adapter
Warp Type-C Cable (Support USB 2.0)
Quick Start Guide
Welcome Letter
Safety Information and Warranty Card
LOGO Sticker
Screen Protector
SIM Tray Ejector

Fig. 40

Source: <https://www.oneplus.com/8/specs>, Page 4, Last Accessed May 18, 2021, Exhibit A

The accused product displays “Warp charging” (i.e., correct category of the charger adapter) as it gets connected (i.e., detecting its availability) with the Warp Charge 30 Power Adapter via the Warp Type-C cable as shown in Fig. 41.

Citation 41: Warp Charging display



Fig. 41

Source: <https://www.androidcentral.com/warp-charge>, Page 3, Last Accessed July 24, 2020, Exhibit B

<p>5b. configuring, by the apparatus, battery charging based in the category of the charger adapter.</p>	<p>The accused product practices a method comprising configuring, by the apparatus, battery charging based in the category of the charger adapter.</p> <p>Once the accused product is connected with the Warp Charge 30 Power Adapter (i.e., charger adapter), the battery charging is configured, and the current battery status is displayed on the screen. See Fig. 42.</p> <p style="text-align: center;">Citation 42: Warp Charging display</p> 
---	---

	<p style="text-align: center;">Fig. 42</p> <p>Source: https://www.androidcentral.com/warp-charge, Page 3, Last Accessed July 24, 2020, Exhibit B</p>
6Pre. The method of claim 1, further comprising: 6a. identifying, by the apparatus, a correct category of the charger adapter after detecting its availability; and	<p>The accused product practices a method identifying, by the apparatus, a correct category of the charger adapter after detecting its availability.</p> <p><i>Refer to supporting evidence of claim element 5[a].</i></p>
6b. using, by the apparatus, the category of the charger adapter to improve accuracy of an initial remaining charging time estimation.	<p>The accused product practices a method comprising using, by the apparatus, the category of the charger adapter to improve accuracy of an initial remaining charging time estimation.</p> <p>In an exemplary scenario, when a Warp Charge 65T (e.g., OnePlus charger adapter provided inbox with the device) is used to charge the OnePlus 9, the OnePlus 9 determines the adapter information (e.g., manufacturer, power details, etc.) to calculate and display the remaining charging time.</p>

In an exemplary scenario, when a general adapter (e.g., not OnePlus adapter) with same power as used in previous scenario (i.e., same power as Warp Charge 65T) is used to charge the OnePlus 9, the OnePlus 9 determines the adapter information (e.g., manufacturer, power details, etc.) to calculate and display the remaining charging time.

Upon information and belief, the estimation of remaining charging time in both the cases are different due to the type of adapter taken into consideration for estimation. See Fig. 43 and Fig. 44.

Citation 43: Warp Charge 65T require 29 Minutes to fully charge OnePlus 9

Warp Charge 65T – Same Wattage but Better Performance

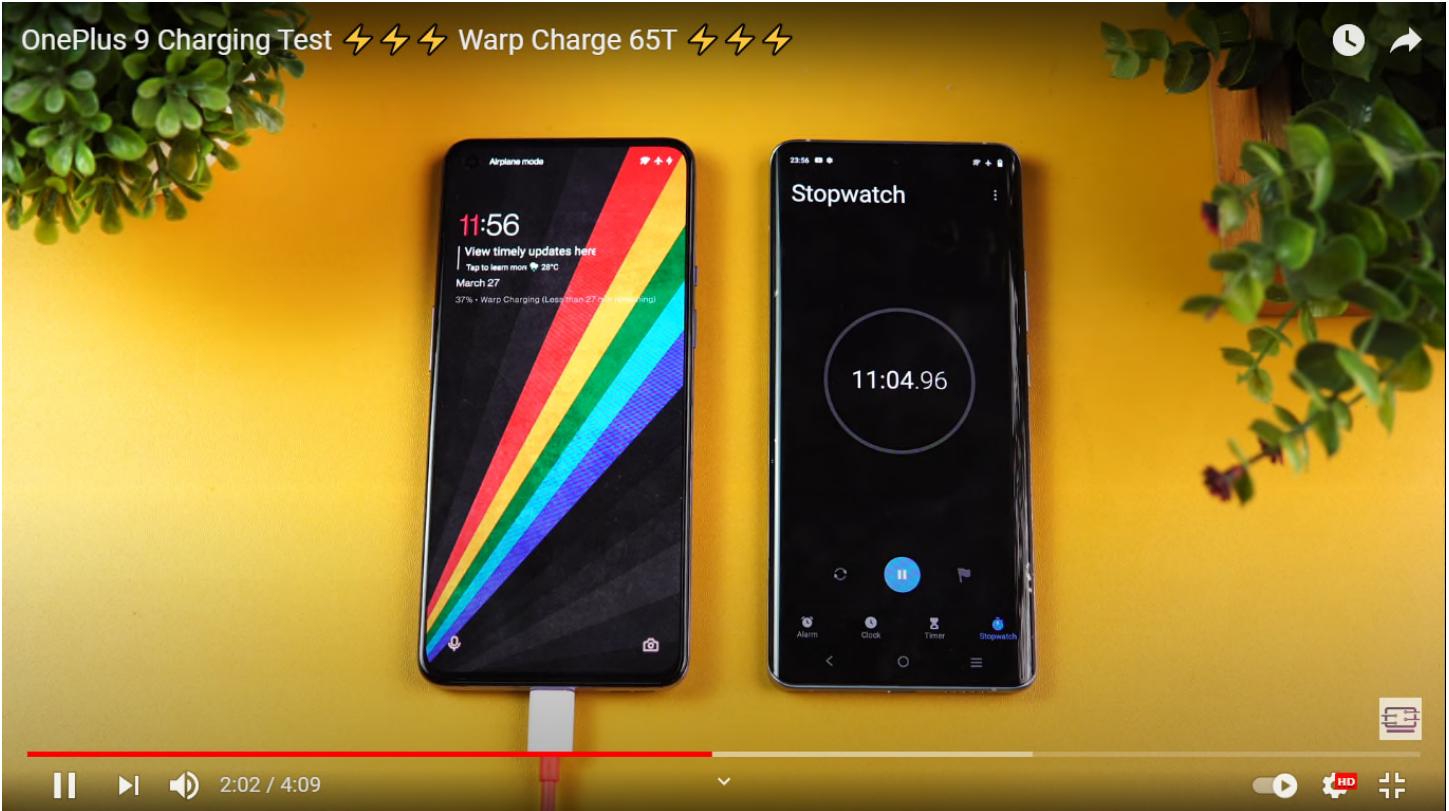
We've also upgraded wired charging on the OnePlus 9 and 9 Pro. We still use the same dual-battery layout present on the OnePlus 8T, with the two internal batteries charging at up to 32.5W, for a combined peak charging of 65W.

What's changed with the upgraded battery is that we've selected a number of upgraded components, in combination with system optimizations. Compared with the OnePlus 8T's Warp Charge 65, the OnePlus 9 and 9 Pro use a brand-new custom battery cell to reduce internal resistance, thereby also reducing battery cell heating during charges. This gives the OnePlus 9 and 9 Pro the ability to charge at a higher output for an extended period so the battery can fill up faster.

While the OnePlus 8T takes 39 minutes to charge from 1% to 100% with Warp Charge 65, the OnePlus 9 and 9 Pro only take **29 minutes to charge from 1% to 100% with Warp Charge 65T**, making it one of the fastest wired charging solutions on the market.

Fig. 43

Source: <https://forums.oneplus.com/threads/deep-dive-lets-talk-charging.1403556/>, Page 3, Last Accessed June 24, 2021, Exhibit L

	<p>Citation 44: OnePlus 9 lock screen display remaining charging time estimate</p>  <p>OnePlus 9 Charging Test ⚡⚡⚡ Warp Charge 65T ⚡⚡⚡</p> <p>11:56 View timely updates here Tap to learn more 28°C March 27 37% • Warp Charging (Less than 27 min charging)</p> <p>Stopwatch 11:04.96</p> <p>2:02 / 4:09</p>
7. The method of claim 1, further comprising:	The accused product practices a method comprising calculating, by the apparatus, a time remaining to charge in the constant current phase based on the constant current phase charging characteristics and a time remaining to

<p>calculating, by the apparatus, a time remaining to charge in the constant current phase based on the constant current phase charging characteristics and a time remaining to charge in the constant voltage phase based on the constant voltage phase charging characteristics, if the battery charging point is in the constant current phase.</p>	<p>charge in the constant voltage phase based on the constant voltage phase charging characteristics, if the battery charging point is in the constant current phase.</p> <p><i>Refer to supporting evidence of claim element 1[c] and 1[e].</i></p>
<p>14Pre. A non-transitory computer readable medium, comprising program instructions, which when executed</p>	<p>The accused product comprises a non-transitory computer readable medium comprising program instructions, which when executed by a computer processor.</p> <p>OnePlus 8 series comprises of 5G supported Qualcomm Snapdragon 865 processor along with RAM/ROM for various storage purposes. See Fig. 45.</p>

by a computer processor, perform:

Citation 45: One Plus 8 Specifications

Performance

Operating System: OxygenOS based on Android™ 10
 CPU: Qualcomm® Snapdragon™865
 5G Chipset: X55
 GPU: Adreno 650
 RAM: 8GB/12GB LPDDR4X
 Storage: 128GB/256GB UFS 3.0 2-LANE
 Battery: 4300 mAh (non-removable)
 Warp Charge 30T Fast Charging (5V/6A)



Fig. 45

Source: <https://www.oneplus.in/8/specs>, Page 1-2, Last Accessed May 18, 2021, Exhibit A

Moreover, the accused product contains program instructions for the fulfillment of various purposes. See Fig. 46.

Citation 46: Exemplary Program Instructions used in the accused product

```

144 static ssize_t power_supply_show_property(struct device *dev,
145                                     struct device_attribute *attr,
146                                     char *buf) {
147     ssize_t ret;
148     struct power_supply *psy = dev_get_drvdata(dev);
149     enum power_supply_property psp = attr->power_supply_attrs;
150     union power_supply_propval value;
  
```

Fig. 46

	<p>Source:</p> <p>https://github.com/OnePlusOSS/android_kernel_oneplus_sm8250/blob/cad09e061ef6cf689a4a1e54d27562e0f042236a/drivers/power/power_supply/power_supply_sysfs.c#L353, Page 2, Last Accessed May 18, 2021, Exhibit E</p>
14a. detecting an availability of a charger adapter;	<p>The accused product comprises a non-transitory computer readable medium comprising program instructions for detecting an availability of a charger adapter.</p> <p><i>Refer to supporting evidence of claim element 1[a].</i></p>
14b. determining whether a battery charging point is in a constant current phase or in a constant voltage phase, based on pre-determined battery charging characteristics, wherein the pre-determined battery charging characteristics comprise constant current phase charging characteristics and constant voltage phase charging characteristics	<p>The accused product comprises a non-transitory computer readable medium comprising program instructions for determining whether a battery charging point is in a constant current phase or in a constant voltage phase, based on pre-determined battery charging characteristics, wherein the pre-determined battery charging characteristics comprise constant current phase charging characteristics and constant voltage phase charging characteristics.</p> <p><i>Refer to supporting evidence of claim element 1[b].</i></p>

and constant voltage phase charging characteristics;	
14c. calculating a time remaining to charge in the constant current phase based on the constant current phase charging characteristics, if the battery charging point is in the constant current phase; and	<p>The accused product comprises a non-transitory computer readable medium comprising program instructions for calculating a time remaining to charge in the constant current phase based on the constant current phase charging characteristics, if the battery charging point is in the constant current phase.</p> <p><i>Refer to supporting evidence of claim element 1[c].</i></p>
14d. calculating a time remaining to charge in the constant voltage phase based on the constant voltage phase charging characteristics, if the battery charging point	<p>The accused product comprises a non-transitory computer readable medium comprising program instructions for calculating a time remaining to charge in the constant voltage phase based on the constant voltage phase charging characteristics, if the battery charging point is in the constant voltage phase.</p> <p><i>Refer to supporting evidence of claim element 1[d].</i></p>

is in the constant voltage phase;	
14e. wherein the time remaining to charge in the constant current phase is based on stored charge characteristics in the constant current phase, when the battery charging point is in the constant current phase and wherein battery stored charge characteristics comprises a battery stored charge value based on monitored tracking of battery charging and discharging.	<p>The accused product comprises a non-transitory computer readable medium comprising program instructions wherein the time remaining to charge in the constant current phase is based on stored charge characteristics in the constant current phase, when the battery charging point is in the constant current phase and wherein battery stored charge characteristics comprises a battery stored charge value based on monitored tracking of battery charging and discharging.</p> <p><i>Refer to supporting evidence of claim element 1[e].</i></p>
15Pre. An apparatus, comprising:	The accused product comprises at least one processor.

<p>15a. at least one processor;</p>	<p>OnePlus 8 series comprises of 5G supported Qualcomm Snapdragon 865 processor. See Fig. 47.</p> <p style="text-align: center;">Citation 47: One Plus 8 Specifications</p> <table border="0" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 25%; vertical-align: top; padding-right: 10px;"> Performance </td><td> Operating System: OxygenOS based on Android™ 10 CPU: Qualcomm® Snapdragon™865 5G Chipset: X55 GPU: Adreno 650 RAM: 8GB/12GB LPDDR4X Storage: 128GB/256GB UFS 3.0 2-LANE Battery: 4300 mAh (non-removable) Warp Charge 30T Fast Charging (5V/6A) </td></tr> </table> <div style="text-align: right; margin-top: 20px;">  </div> <p style="text-align: center;">Fig. 47</p> <p>Source: https://www.oneplus.in/8/specs, Page 1-2, Last Accessed May 18, 2021, Exhibit A</p>	Performance	Operating System: OxygenOS based on Android™ 10 CPU: Qualcomm® Snapdragon™865 5G Chipset: X55 GPU: Adreno 650 RAM: 8GB/12GB LPDDR4X Storage: 128GB/256GB UFS 3.0 2-LANE Battery: 4300 mAh (non-removable) Warp Charge 30T Fast Charging (5V/6A)
Performance	Operating System: OxygenOS based on Android™ 10 CPU: Qualcomm® Snapdragon™865 5G Chipset: X55 GPU: Adreno 650 RAM: 8GB/12GB LPDDR4X Storage: 128GB/256GB UFS 3.0 2-LANE Battery: 4300 mAh (non-removable) Warp Charge 30T Fast Charging (5V/6A)		
<p>15b. at least one memory including computer program code;</p>	<p>The accused product comprises at least one memory including computer program code.</p> <p>OnePlus 8 comprises RAM and ROM for various storage purposes. See Fig. 48.</p> <p style="text-align: center;">Citation 48: One Plus 8 Specifications</p>		

	<p>Performance</p> <p>Operating System: OxygenOS based on Android™ 10 CPU: Qualcomm® Snapdragon™865 5G Chipset: X55 GPU: Adreno 650 RAM: 8GB/12GB LPDDR4X Storage: 128GB/256GB UFS 3.0 2-LANE Battery: 4300 mAh (non-removable) Warp Charge 30T Fast Charging (5V/6A)</p> 
15c. the at least one memory and the computer program code configured to, with the at least one processor, cause the processor at least to:	<p>The accused product comprises at least one memory and the computer program code configured with the at least one processor.</p> <p>OnePlus 8 series comprises of 5G supported Qualcomm Snapdragon 865 processor along with RAM/ROM for various storage purposes. See Fig. 49.</p> <p style="text-align: right;">Citation 49: One Plus 8 Specifications</p>

Performance

Operating System: OxygenOS based on Android™ 10
CPU: Qualcomm® Snapdragon™865
5G Chipset: X55
GPU: Adreno 650
RAM: 8GB/12GB LPDDR4X
Storage: 128GB/256GB UFS 3.0 2-LANE
Battery: 4300 mAh (non-removable)
Warp Charge 30T Fast Charging (5V/6A)



Fig. 49

Source: <https://www.oneplus.in/8/specs>, Page 1-2, Last Accessed May 18, 2021, Exhibit A

Moreover, the accused product contains computer program code for the fulfillment of various purposes. See Fig. 50.

Citation 50: Exemplary Computer Program Code used in the accused product

```
144 static ssize_t power_supply_show_property(struct device *dev,  
145                                     struct device_attribute *attr,  
146                                     char *buf) {  
147     ssize_t ret;  
148     struct power_supply *psy = dev_get_drvdata(dev);  
149     enum power_supply_property psp = attr->power_supply_attrs;  
150     union power_supply_propval value;
```

Fig. 50

	<p>Source:</p> <p>https://github.com/OnePlusOSS/android_kernel_oneplus_sm8250/blob/cad09e061ef6cf689a4a1e54d27562e0f042236a/drivers/power/power_supply/power_supply_sysfs.c#L353, Page 2, Last Accessed May 18, 2021, Exhibit E</p>
15d. detect an availability of a charger adapter;	<p>The accused product is an apparatus comprising at least one processor to detect an availability of a charger adapter.</p> <p><i>Refer to supporting evidence of claim element 1[a].</i></p>
15e. determine whether a battery charging point is in a constant current phase or in a constant voltage phase, based on pre-determined battery charging characteristics, wherein the pre-determined battery charging characteristics comprise constant current phase charging characteristics and constant voltage	<p>The accused product is an apparatus comprising at least one processor to determine whether a battery charging point is in a constant current phase or in a constant voltage phase, based on pre-determined battery charging characteristics, wherein the pre-determined battery charging characteristics comprise constant current phase charging characteristics and constant voltage phase charging characteristics.</p> <p><i>Refer to supporting evidence of claim element 1[b].</i></p>

phase charging characteristics;	
15f. calculate a time remaining to charge in the constant current phase based on the constant current phase charging characteristics, if the battery charging point is in the constant current phase; and	<p>The accused product is an apparatus comprising at least one processor to calculate a time remaining to charge in the constant current phase based on the constant current phase charging characteristics, if the battery charging point is in the constant current phase.</p> <p><i>Refer to supporting evidence of claim element 1[c].</i></p>
15g. calculate a time remaining to charge in the constant voltage phase based on the constant voltage phase charging characteristics, if the battery charging point is in the constant voltage phase;	<p>The accused product is an apparatus comprising at least one processor to calculate a time remaining to charge in the constant voltage phase based on the constant voltage phase charging characteristics, if the battery charging point is in the constant voltage phase.</p> <p><i>Refer to supporting evidence of claim element 1[d].</i></p>

<p>15h. wherein the time remaining to charge in the constant current phase is based on stored charge characteristics in the constant current phase, when the battery charging point is in the constant current phase and wherein battery stored charge characteristics comprises a battery stored charge value based on monitored tracking of battery charging and discharging.</p>	<p>The accused product is an apparatus comprising at least one processor wherein the time remaining to charge in the constant current phase is based on stored charge characteristics in the constant current phase, when the battery charging point is in the constant current phase and wherein battery stored charge characteristics comprises a battery stored charge value based on monitored tracking of battery charging and discharging.</p> <p><i>Refer to supporting evidence of claim element 1[e].</i></p>
<p>16. The apparatus of claim 15, wherein the battery stored charge characteristics</p>	<p>The accused product is an apparatus comprising at least one processor wherein the battery stored charge characteristics comprise data that provides a remaining charging time estimate based on a battery stored charge estimate.</p>

comprise data that provides a remaining charging time estimate based on a battery stored charge estimate.	<i>Refer to supporting evidence of claim element 2.</i>
---	---